TOEPASSINGEN VOOR MSX-COMPUTERS

met besturingsprojecten



Max Voorburg

Kluwer Software-reeks

Toepassingen voor MSX-computers

Hoofdstuk Besturingsprogramma's

Max Voorburg

Kluwer 1987

Scanned, ocr'ed and converted to PDF by HansO, 2001

Inhoud

4. Besturingsprogramma's

Het zal nauwelijks toelichting behoeven dat computers erg belangrijk zijn geworden bij het besturen van processen in velerlei takken van industrie. Robots die worden gebruikt voor de assemblage van auto's, zijn voorzien van computers voorde besturing, zelfs complete rangeerheuvels worden door computers bestuurd, maar ook liften. In dit gedeelte van het boek kunt u kennis maken met besturing door de computer, waarbij uw MSX-computer het hart vormt van het totale systeem. U kunt zich voorstellen dat dit niet zomaar kan.

De MSX-computer moet worden uitgebreid met een elektronische schakeling, die dit mogelijk maakt. Zo'n elektronische schakeling wordt interface genoemd. Deze aanduiding zal verder in de tekst worden gebruikt. De interface bestaat uit acht relais met een voedingsgedeelte. De aansturing van de interface geschiedt via de pnnterpoort van de MSX-computer. Dit heeft het grote voordeel dat de aansturing rechtstreeks vanuit BASIC kan. Velen van u zullen zich afvragen waarom voor relais is gekozen. De keuze is daarop gevallen omdat relais tal van spanningen kunnen doorschakelen en de werking ervan voor iedereen snel te begrijpen is. De opbouw van de interface is vrij eenvoudig, ook voor diegenen onder u die nog nooit hebben gesoldeerd. Met goed gereedschap en nauwkeurig werken kunt u dit project zeker tot een goed einde brengen.
Het benodigde gereedschap bestaat uit een soldeerbout van 15 watt, een tangetje, een striptang en harskernsoldeer.

Er worden twee besturingsprojecten als voorbeeld behandeld. Dit zijn een wagentje, een zogenaamde schildpad en een stuk treinbaan om de werking van sensors uit te leggen. Bij de ontwikkeling van de experimenten stond voorop dat de modellen snel en makkelijk moesten worden nagebouwd. LEGO biedt die mogelijkheid. Daarbij komt het grote voordeel dat haast iedereen LEGO kent en men vaak over voldoende LEGO beschikt om de modellen na te bouwen. Uiteraard is het niet noodzakelijk om LEGO te gebruiken; de experimenten kunnen ook met bijvoorbeeld Fisher Technik worden uitgevoerd. In dit hoofdstuk zijn we echter van LEGO uitgegaan. Eén onderdeel in de projecten komt niet in de LEGO-collectie voor: het glas-schakelaartje, dat u in de elektronicawinkel kunt kopen. Met behulp van foto's en tekeningen is getracht u een goed beeld te geven hoe de modellen in elkaar zitten. Daarnaast is het de bedoeling dat deze projecten als basis dienen voor uw eigen experimenten. Wat dacht u van een robotarm, een volautomatische containerkraan of een eenvoudige computergestuurde rangeerheuvel? Ideeën genoeg, dacht ik zo.

Bij de ontwikkeling van de interface is van een zestal basisideeën uitgegaan:

- De interface moet eenvoudig te bouwen zijn.
- De gebruikte onderdelen moeten gemakkelijk te verkrijgen zijn.
- De interface moet uitnodigen zelf te gaan experimenteren. Ook voor mensen die weinig of geen elektronische kennis hebben
- De softwarematige aansturing geschiedt in BASIC.
- De koppeling met de MSX-computer mag geen enkel probleem veroorzaken.

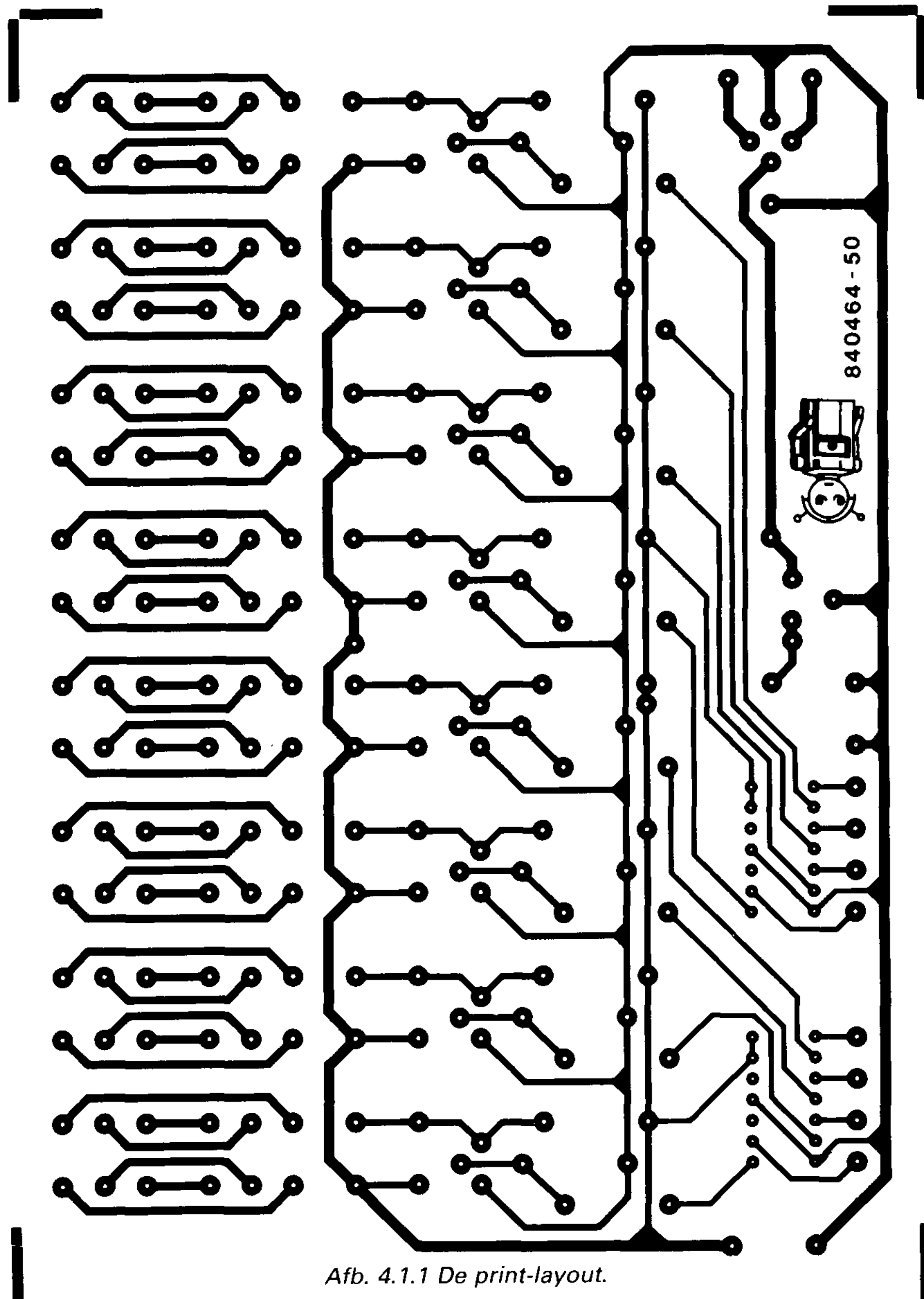
4.1 De bouw van de interface

De gebruikte schakeling is zeer simpel en u kunt het principe in elk elektronica-boek terugvinden. Via de printerpoort wordt een signaal aangeboden aan de interface. Dit signaal ligt op TTL-niveau. Dat wil zeggen dat vijf volt een logische één geeft en dat nul volt een logische nul voorstelt. Het signaal van de computer wordt opgevangen door twee IC's, zogenaamde hexinverters. Deze IC's doen dienst als buffers. Mocht er zich onverhoopt kortsluiting op de interface voordoen, dan zorgen deze twee buffers ervoor dat er niets met de computer gebeurt. Na deze buffers komt een transistorversterkingstrap. Deze versterkingstrap is nodig om het relais te kunnen aansturen. Het signaal uit de buffers is te zwak om rechtstreeks vanuit de buffer-IC's de relais aan te sturen. Op de printkaart is de interfaceschakeling met de voedingsschakeling geïntegreerd. Ook dit voedingsgedeelte is een standaardoplossing zonder afregel-componenten. U kunt de schakeling voeden met zowel wisselspanning als gelijkspanning. De voedingsspanning moet liggen tussen de zes en tien volt en moet zo'n 400 mA kunnen leveren. Bij het aansluiten van de spanning hoeft u in geval van gelijkspanning niet te letten op de plus- en de minaansluiting. Eventueel kunt u de interface ook voeden met twee platte, in serie geschakelde 4,5 volt batterijen. Een nadeel daarvan iswel dat ze tamelijk snel leeg raken.

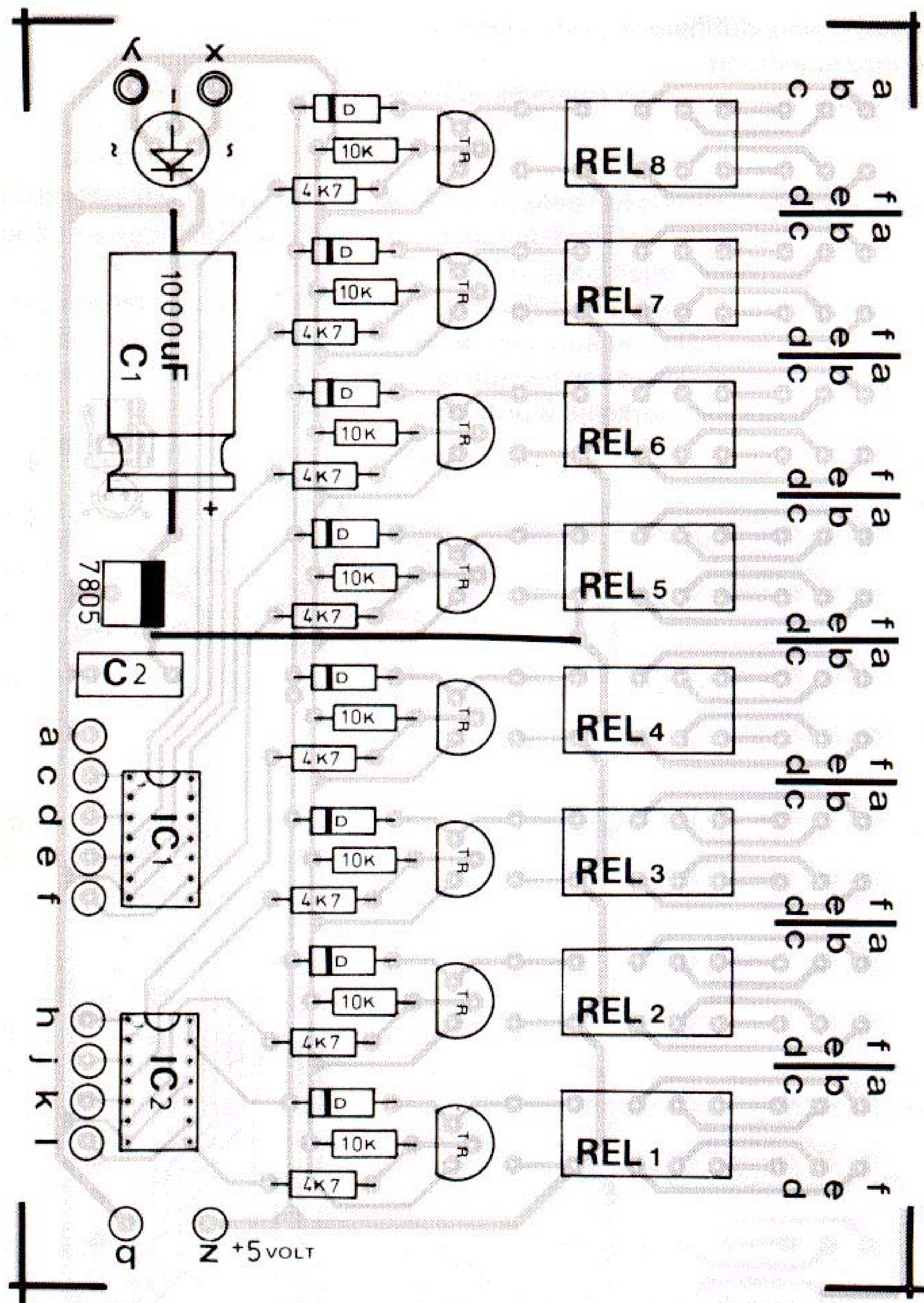
Om de interface te kunnen bouwen, heeft u de volgende onderdelen nodig:

1 printkaart zoals afgebeeld in afbeelding 4.1.1.
1 printerconnector Centronicstype, 14 pens Amphenol.
2 IC's 7404 of 74LS04 of 7414 of 74LS14 of 40106.
8 transistors BC 108 of BC 547.
8 dioden 1 N914 of 1N4001.
8 weerstanden 4,7 kilo-ohm.
8 weerstanden 10 kilo-ohm.
1 bruggelijkrichter B 40 C 1500.
1 elco 1000 uF/40V axiaal.
1 spanningsstabilisator 7805.
1 condensator 0,1 |uF met een steek van 10 mm.
1 zakje soldeerpennen met een doorsnede van 1,3 mm.
1 zakje met 100 bijbehorende soldeerklemmen voor bovengenoemde pennen.
8 miniatuurrelais dubbelpolig om 5 of 6 volt.
1 stuk montagedraad.
1 stuk tienaderige kabel van dertig centimeter lengte.

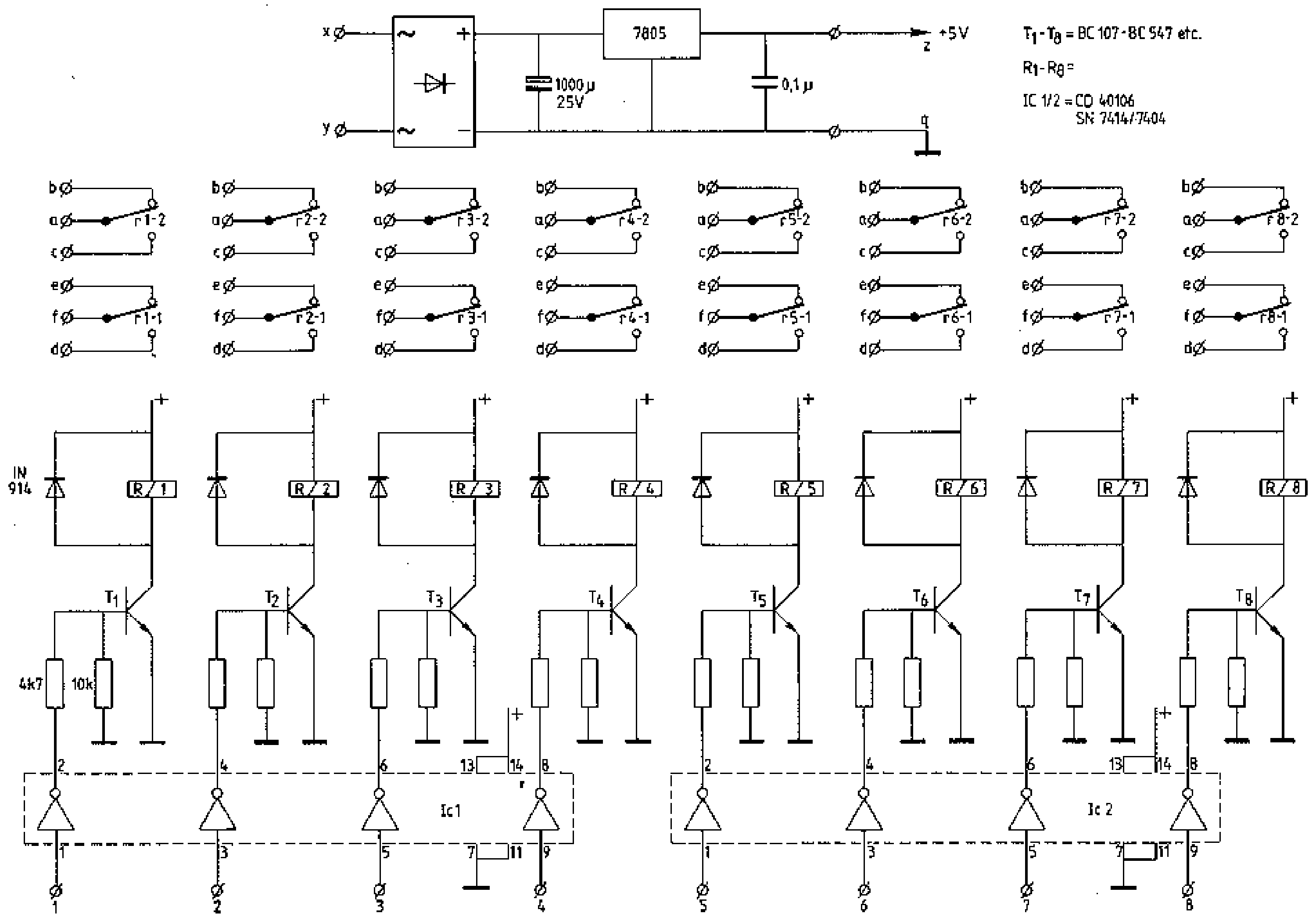
Het is het handigste om bij het aanschaffen van de relais de printkaart mee te nemen. U kunt dan controleren of de relais passen, hoewel de meeste dubbel-polige miniatuurrelais zullen passen.



Afb. 4.1.1 De print-layout.



REL 8
REL 7
REL 6
REL 5
REL 4
REL 3
REL 2
REL 1
D
10K
4K7
TR
1000uF
C1
C2
7805
IC1
IC2
+5 VOLT

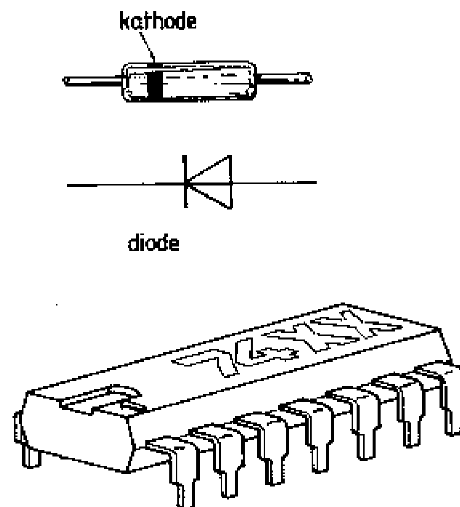


Afb. 4.1.3 Het elektronisch schema.

Bouwinstructie

U legt de printkaart met componentopdruk naar boven.

1. Als eerste soldeert u de doorverbinding die begint tussen relais 4 en 5 en eindigt tussen de 7805 en C2 (zie afbeelding 4.1.2). Deze doorverbinding maakt u met behulp van een stuk montagedraad.
2. U soldeert nu de soldeerpennen in de gaten X,Y en in de gaten a,b,c,d,e en f boven de acht relais. De letteraanduiding vindt u op de componenten-layout (afbeelding 4.2.2) De combinatie van acht groepen van zes soldeerpennen wordt het 'speldenkussen' genoemd. Meer pennen hoeven niet te worden gesoldeerd.
3. U buigt nu de weerstanden van 10 kilo-ohm op maat. De kleurcode is bruin-zwart-oranje. Hierna soldeert u deze 10 kilo-ohm-weerstanden op de daarvoor bestemde plaats. Deze is aangegeven op de printkaart (zie afbeelding 4.1.2).
4. Hetzelfde doet u met de weerstanden van 4,7 kilo-ohm. De kleurcode van deze weerstanden is geel-violet-rood.
5. Nu komen de acht dioden aan de beurt. Deze zijn te herkennen aan een rode of witte ring, zie afbeelding 4.1.4. Dezelfde aanduiding treft u op de printkaart aan, zie afb 4.1.2. U moet er op letten dat de diode in de goede richting wordt gemonteerd. De ring van de diode moet naar beneden wijzen.



Afb. 4.1.4

6. Op de printkaart worden de plaatsen van de transistor aangeduid met TR. Tevens is het bovenaanzicht van de transistor aangegeven. U plaatst de transistors zodanig dat het bovenaanzicht overeenkomt met de opdruk op de printkaart, zie afb 4.1.1. U hoeft de pootjes van de transistor slechts een klein beetje te verbuigen.
7. Het is verstandig de IC's niet direct op de printkaart te solderen. Inplaats hiervan kunt u beter veertienpens IC-voetjes solderen. In deze voetjes kunt u dan de IC's drukken. Dit vastsolderen moet u zeer nauwkeurig doen; het blijft een precisiewerkje. U moet

erop letten dat de IC's in de goede richting in de voetjes worden gedrukt. Op de componenten-layout (zie afb 4.1.2) is het bovenaanzicht van de IC's aangegeven.

8. Op de bruggelijkrichter staat aangegeven wat de wisselstroomingangen zijn en wat de gelijkspanningsuitgangen zijn. U zorgt ervoor dat de wisselspanningingangen als het ware worden verbonden met de punten X en Y. De pluskant van de bruggelijkrichter wijst naar de elco. Op de printkaart zijn de twee wisselspanningsingangen aangegeven met een slangetje en de min-kant van de gelijkrichter is eveneens aangegeven. Als u zeker weet dat de bruggelijkrichter goed zit, soldeert u deze vast.
9. De pluszijde van de grote axiale elco heeft aan één zijde een groef, een zwarte of witte ring of een plusteken.Wanneerde print voor u ligt met het 'speldenkussen' zover mogelijk van u vandaan, moet de pluszijde van de elco naar rechts wijzen. Zo soldeert u hem vast.
10. De spanningsstabilisator 7805 kan maar op één manier op de print worden gesoldeerd. Het kleine metalen koelplaatje moet in de richting van de weerstanden wijzen. Het tekeningetje op de printkaart biedt voldoende houvast.
11. Tenslotte moet de condensator van 0,1 uF worden gemonteerd. Deze komt op de plaats C2 tussen de IC's en de 7805 spanningsstabilisator.
12. Tenslotte kunnen de acht relais worden vastgesoldeerd. Als u de goede relais heeft aangeschaft, kunnen deze maar op één manier op de printkaart worden gesoldeerd.
13. Als laatste komt het vervelendste klusje: de verbinding tussen printkaart en connector.Nu is de gehele interface gereed. Voordat hij met de computer wordt verbonden, moet u nauwkeurig controleren of alle onderdelen op de goede plaats en in de goede richting gemonteerd zijn en of de kabel van de interface op de juiste manier aan de connector is verbonden. Zonder de interface aan de computer te verbinden, sluit u een spanning tussen 6 en 10 volt aan op de punten Xen Y. Met een universeelmeter, ingesteld op gelijkspanning, meet u de spanning op de punten Z en Q. Deze spanning moet precies 5 volt zijn. De genoemde punten vindt u op de componenten-layout.

PEN	NAAM	I/O	PEN AANSLUITING
1	$\overline{PSTB}$	O	
2	PDB0	O	
3	PDB1	O	
4	PDB2	O	
5	PDB3	O	
6	PDB4	O	
7	PDB5	O	
8	PDB6	O	
9	PDB7	O	
10	N.C.	–	
11	BUSY	I	
12	N.C.	–	
13	N.C.	–	
14	GND	–	

Afb. 4.1.5 De 14 pens-connector, vanaf de achterkant gezien.

Centronics-connector	*verbinden met*	*Interface-kaart*	*Omschrijving*
pen 2	. . . .	gat c	databit 0
pen 3	. . . .	gat d	databit 1
pen 4	. . . .	gat e	databit 2
pen 5	. . . .	gat f	databit 3
pen 6	. . . .	gat h	databit 4
pen 7	. . . .	gat j	databit 5
pen 8	. . . .	gat k	databit 6
pen 9	. . . .	gat l	databit 7
pen 11	. . . .	gat q	strobe
pen 14	. . . .	gat a	aarde

4.2 Proef-interface

Dit programma vormt de basis voor alle hierna te bespreken besturingsprogramma's. In dit programma zijn alle bouwstenen terug te vinden die nodig zijn om eigen besturingsprogramma's te ontwikkelen. Tevens biedt het programma u de gelegenheid de werking van de LEGO-modellen te beproeven. Ik wijs u erop dat het programma alleen werkt als de interface aangesloten is op de computer, via de printerpoort. Bij het overtikken van het programma moet u erop letten dat in de module 'Interface' bij het LPRINT-statement achter de functie CHR$(B) de puntkomma niet wordt vergeten. Vergeet u deze puntkomma dan zal de interface niet werken.

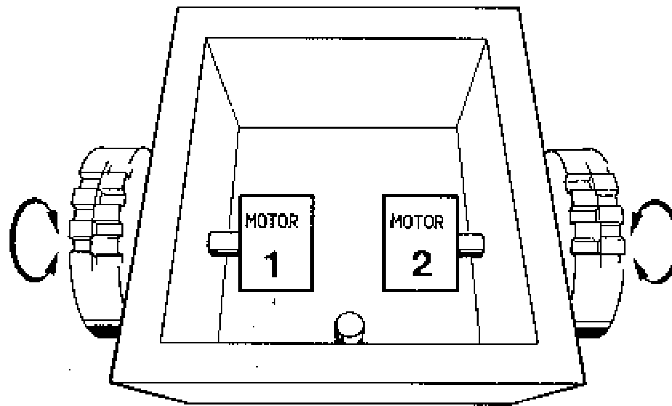
De interface wordt via BASIC aangestuurd door middel van de CHR$(B)-functie. De variabele B mag de waarde O t/m 255 hebben. Elk getal tussen O en 255 bepaalt welke relais zijn aangetrokken en welke niet. Het zou logisch zijn om met het decimale getal nul alle relais in de ruststand te zetten. Feitelijk gebeurt dit ook, maar op de interface zitten twee inverter-IC's. Deze IC's, die als buffer fungeren, maken van een binaire één een binaire nul en van een binaire nul een binaire één. De decimale nul omgezet naar binair geeft 0000 0000. Na de inverter-IC's wordt dit 1111 1111. Om de relais in ruststand te brengen, moet er 1111 1111 naar de interface worden gestuurd. Dit komt overeen met 255 decimaal. Wilt u bijvoorbeeld het laatste relais activeren dan moet u de waarde 254 naar de interface sturen. 254 decimaal is 1111 1110 binair. Na de inverters op de interface is dit het getal 0000 0001 geworden. Met het programma 'Proef-interface' kunt u elk gewenst bitpatroon naar de interface sturen en daarmee één of meer relais activeren. Dit doet u door de gewenste decimale waarde via het toetsenbord in te voeren. U krijgt de decimale waarde omgezet in de binaire waarde te zien en binair geinverteerd. Met de functietoets F5 stuurt u in één keer de waarde 255 naar de printerpoort. Met de functietoets F1 verlaat u het programma.

```
10 REM *** PROEF INTERFACE ***
20 DIM A(7,2)
30 KEY OFF
40 GOTO 630
50 :
60 REM *** INITIALISEREN ***
70 B = 255
80 H$=STRING$(37,"-")
90 KEY 1,"999"+CHR$(13)
100 KEY5,"255"+CHR$(13)
110 RETURN
120 :
130 REM *** KOP ***
140 CLS         :REM scherm wissen
150 PRINT "*** Proef interface ***
160 RETURN
170 :
180 REM *** UITVOER ***
190 GQSUB 140   :REM *** KOP
200 LOCATE 2,5
210 PRINT "Ingegeven waarde..:";B
220 FOR J = 1 TO 2
230 LOCATE 2,5+J*2
240 A$ = "Bitpatroon........:"
```

```
250 IF J = 1 THEN 270
260 A* = "Ge-inverteerd.....:"
270 PRINT A$;
280 FOR I = 7 TO 0 STEP -1
290 PRINT CHR$(A(I,J)+48);
300 IF I MOD 2 = 0 THEN PRINT " ";
310 NEXT I
320 NEXT J
330 GOSUB 590   :REM *** INTERFACE
340 RETURN
350 :
360 REM *** BITPATROON ***
370 W = B   :REM   onthouden
380 FOR I = 7 TO 0 STEP -1
390 A(I,1) = O : A(I,2) = 1 : R = 2^I
400 IF W-R < 0 THEN 420
410 A(I,1)=1 : A(I,2)= 0 : W=W-R
420 NEXT I
430 RETURN
440 :
450 REM *** INVOER ***
460 LOCATE 0,18 : PRINT H$
470 LOCATE 2,20
480 PRINT "[F1] =stoppen"
490 PRINT "  [F5] = 255"
500 LOCATE 2, 14
510 PRINT "Geef decimale waarde ";
520 PRINT "
530 LOCATE 24,14 : INPUT B
540 IF B = 999 THEN 560
550 IF B < 0 OR B > 255 THEN 500
560 RETURN
570 :
580 PRINT *** INTERFACE ***
590 LPRINT CHR$(B); : REM puntkomma!
600 RETURN
610 :
620 REM *** STUURMODULE ***
630 GOSUB 70    :REM *** INIT
640 GOSUB 370   :REM *** BITPATROON
650 GOSUB 190   :REM *** UITVOER
660 GOSUB 460   :REM *** INVOER
670 IF B <> 999 THEN 640
680 CLS
690 END
```

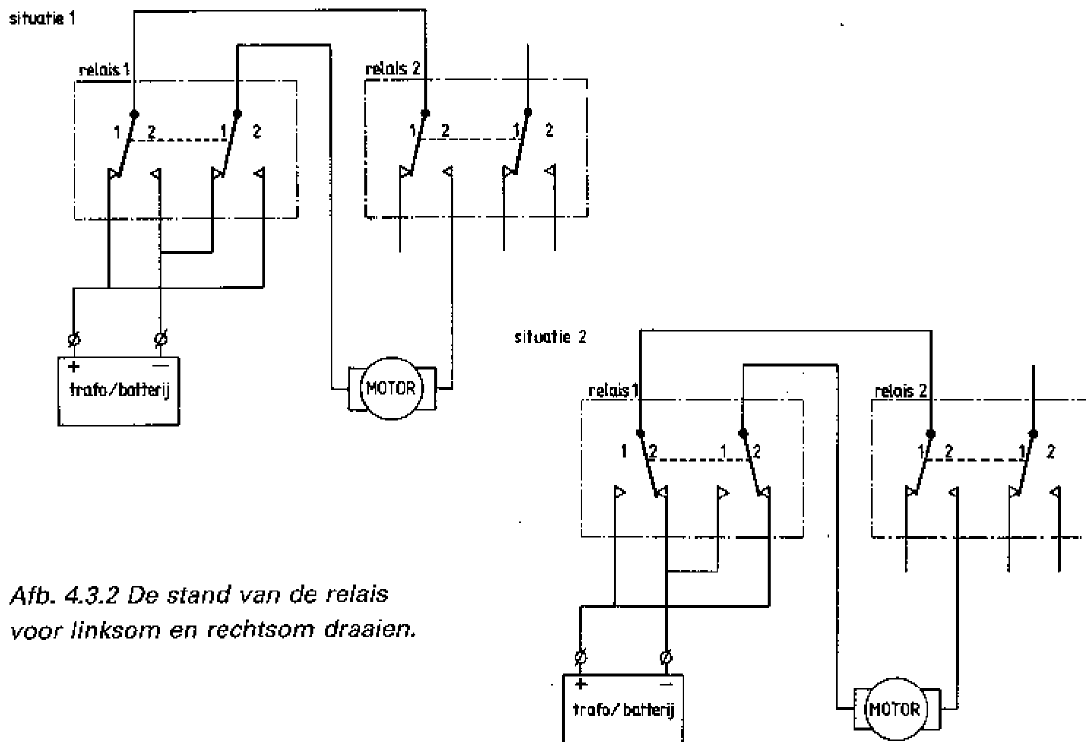
4.3 Schildpad

De werking van de schildpad is gebaseerd op het principe van een rupsvoertuig. Onafhankelijke aandrijving van de wielen, zodat hiermee ook van richting kan worden veranderd. In afbeelding 4.3.1 ziet u het principe. Op een platform zijn twee motoren gemonteerd. Deze motoren drijven ieder een wiel aan. Draaien de motoren beide in voorwaartse richting, dan gaat het wagentje vooruit rijden. Draaien de motoren in achterwaartse richting dan rijdt hetwagentje achteruit. Wanneer de rechtermotor naar voren loopt en de linkermotor naar achteren, zal het wagentje tegen de klok in draaien. Draait de rechtermotor naar achteren en de linkermotor naar voren dan zal het wagentje met de klok meedraaien.



Afb. 4.3.1 Het principe van de schildpad.

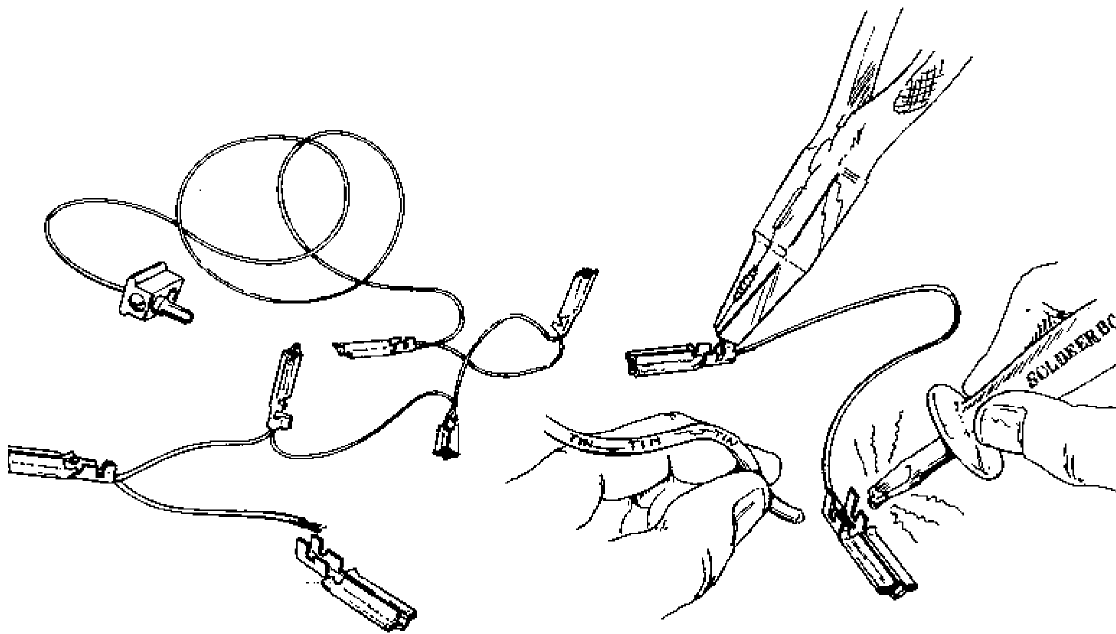
Aan de hand van de fotoserie en de exploded view tekening kunt u zien hoe het wagentje is opgebouwd. Uit de foto's en tekeningen blijkt dat het wagentje feitelijk op twee wielen rijdt en het derde, kleine, wieltje uitsluitend dient om het wagentje in evenwicht te houden. Het wagentje rijdt het beste op een glad oppervlak. Elke motor moet afzonderlijk op commando kunnen stoppen, linksom draaien en rechtsom draaien. Het linksom en rechtsom draaien van een gelijkstroom-motor geschiedt door plus en min van de voedingsspanning naar de motor met elkaar te verwisselen. Verbreekt u de verbinding van de voeding met de motor dan stopt hij. Per motor zijn twee relais nodig om de bovengenoemde functies uit te voeren. Een relais zorgt voor de omschakeling van de spanning waardoor de motor linksom of rechtsom gaat draaien en het tweede relais zorgt voor het aan- en uitzetten van de motor. In afbeelding 4.3.2 kunt u zien hoe dit schematisch in elkaar zit. Relais 1 zorgt voor de omschakeling, relais 2 zorgt voor het in- en uitschakelen van de motor. In het schema staat relais 2 in ruststand, dat betekent dat er geen spanning naar de motor gaat.



Afb. 4.3.2 De stand van de relais voor linksom en rechtsom draaien.

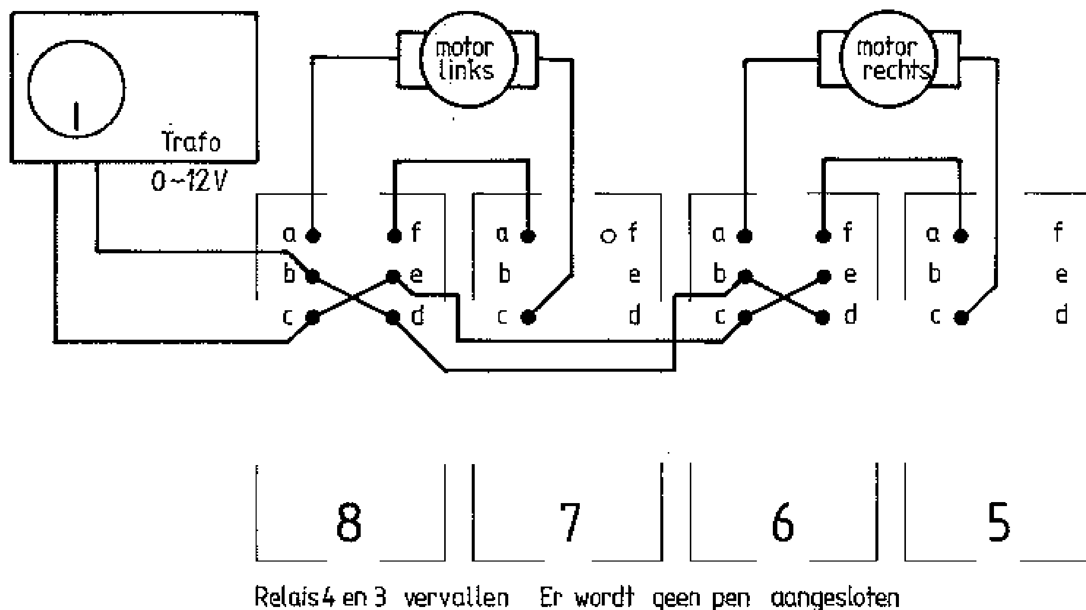
Zodra via de computer relais 2 opdracht krijgt aan te trekken, klapt relais 2 om en is er spanning. De motor gaat draaien. Krijgt relais 1 opdracht om aan te trekken dan worden de plus en de min omgedraaid. In afbeelding 4.3.2 kunt u dat goed zien. Relais 8 en 7 (zie componenten-layout) besturen aandrijfmotor 1. Relais 6 en 5 besturen motor 2. De even genummerde relais zorgen voor het omschakelen van de spanning. De oneven genummerde relais schakelen de motoren in of uit.

Bij de ontwikkeling van de interface is ervan uitgegaan dat de voeding van de interfacekaart en de voeding van de motoren gescheiden zijn. De voeding van de motoren kan geschieden door batterijen of door een transformator. U zult nu eerst een aantal kabeltjes moeten maken, die op het speldenkussen worden geprikt om de verschillende verbindingen te maken tussen de relais onderling en de motoren. Het gemakkelijkste is het om twee verschillende kleuren soepel montagedraad te gebruiken. U knipt van elke kleur draad een stuk van zo'n 25 centimeter. Eveneens van elke kleur vijf stukken van vijf è zes centimeter. De lengte hoeft niet precies te zijn.



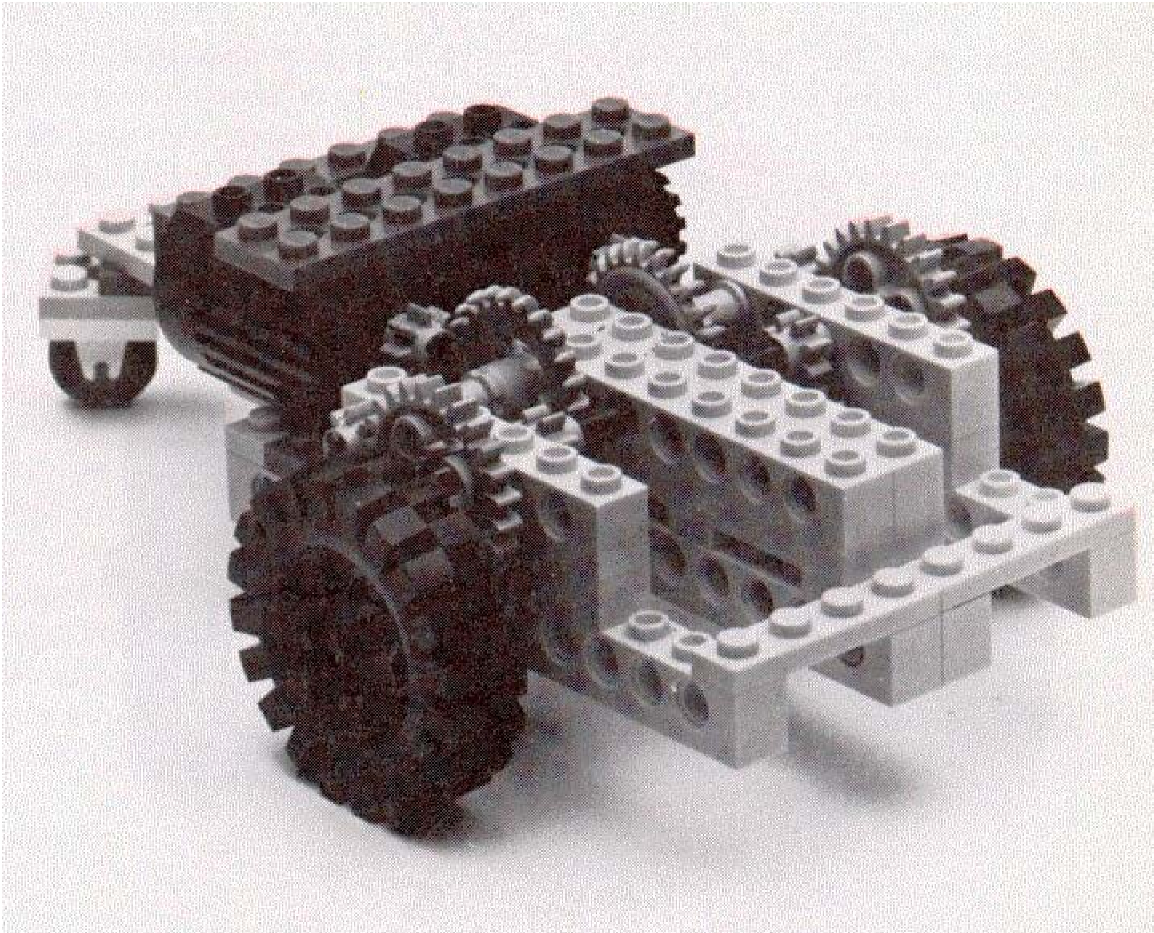
Afb. 4.3.3

Aan de hand van de linkertekening kunt u zien hoe u de twee kabeltjes moet maken. U stript van elk draadje over een afstand van een halve centimeter de isolatie en soldeert deze zoals in afbeelding 4.3.3. U bevestigt ook een LEGO-stekkertje aan elk kabeltje (zie afbeelding 4.3.4). Mocht u geen LEGO-transformator gebruiken dan kunt u dit weglaten. Behalve de twee lange kabeltjes, heeft u nog drie korte kabeltjes nodig met aan elke kant een soldeerklemmetje. Zo'n kort kabeltje is op de tekening rechts afgebeeld. Voor de verbinding met het wagentje heeft u een soepele vieraderige kabel van 2 a 3 meter lengte nodig. Aan de ene kant van de kabel soldeert u aan elke ader afzonderlijk een soldeerklemmetje, aan de andere kant komen vier LEGO-stekkertjes. U moet er op letten dat u kabel gebruikt waarvan de aders verschillende kleuren hebben.



Afb. 4.3.4 Bedradingsschema van het speldenkussen.

In afb.4.3.4 kunt u zien, hoe de relais onderling met de motor en met de voedingsbron worden verbonden. De bedrading is als volgt: u begint met het kabeltje, zoals dat op de tekening links is afgebeeld. Het LEGO-stekkertje steekt u in de transformator of in het batterijhuis. Het eerste klemmetje prikt u op soldeerpen 8B. Het volgende klemmetje prikt u op pen 8D. Het derde klemmetje van de kabel prikt u op pen 6B, het vierde klemmetje gaat op pen 6D. Hetzelfde herhaalt u voordetweede kabel die u heeft gemaakt. Deze prikt u achtereenvolgens op de pennen 8C, 8E, 6C en 6D. De twee korte kabeltjes met aan elke kant een soldeerklemmetje gebruikt u om de verbindingen 8F-7A en 6F-5A te maken. Als laatste kunnen nu de verbindingen met de motoren worden gemaakt. Hiertoe gebruikt u de vieraderige kabel. Aan de hand van de tekening kunt u zien hoe die verbinding tot stand wordt gebracht. De verbinding van de motor links verloopt via pen 8A en 7C. Motor rechts wordt verbonden met pen 6A en 5C.



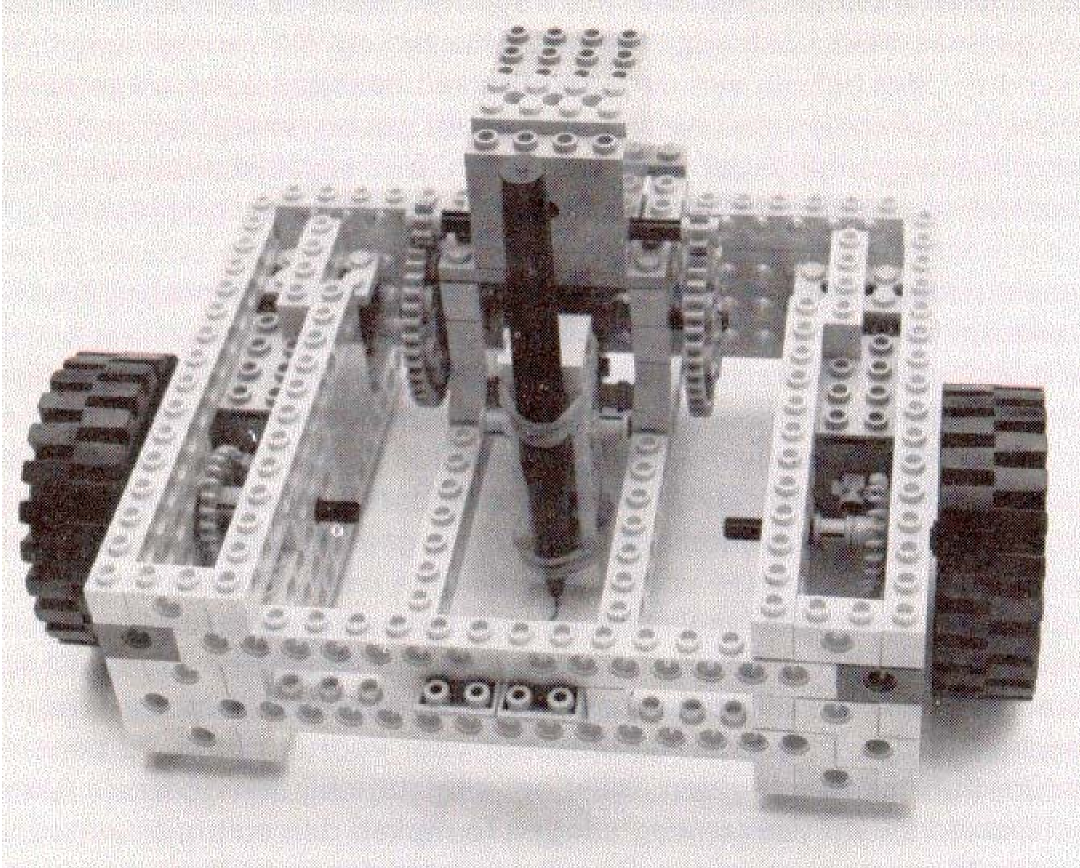
Nu de interface met het wagentje en de transformator of batterijen is verbonden, kan de stekker van de interface in de aansluiting van de printer op de MSX-computer worden gestoken. Tegelijkertijd kunt u ook de voeding van de interface inschakelen door een spanning van 6 tot 10 volt op de punten X en Y van de interface te zetten. Om te controleren of het wagentje goed is aangesloten, laadt u het programma 'Proef-interface'. Het kan zijn dat u de stekkertjes op de motor moet omwisselen. Op het beeldscherm verschijnt 'Geef decimale waarde'. Door ingave van het getal 255 worden alle relais in de ruststand gezet. Toets nu het getal 247 in. De rechtermotor van de schildpad moet nu vooruit gaan draaien. Draait de motor achteruit, verwissel dan de stekkertjes op de motor. Draait de motor in het geheel niet, kijk dan de bedrading na. Hetzelfde moet ook worden gedaan voorde linkermotor. Toets het getal 253 in, waarna de linkermotor vooruit moet gaan draaien. Draait de motor andersom, verwissel dan de stekkertjes op de linkermotor.

De test is nu klaar en u kunt beginnen met de experimenten met het wagentje. Na starten van het programma 'Besturing Schildpad' verschijnt een menu. U heeft drie mogelijkheden:

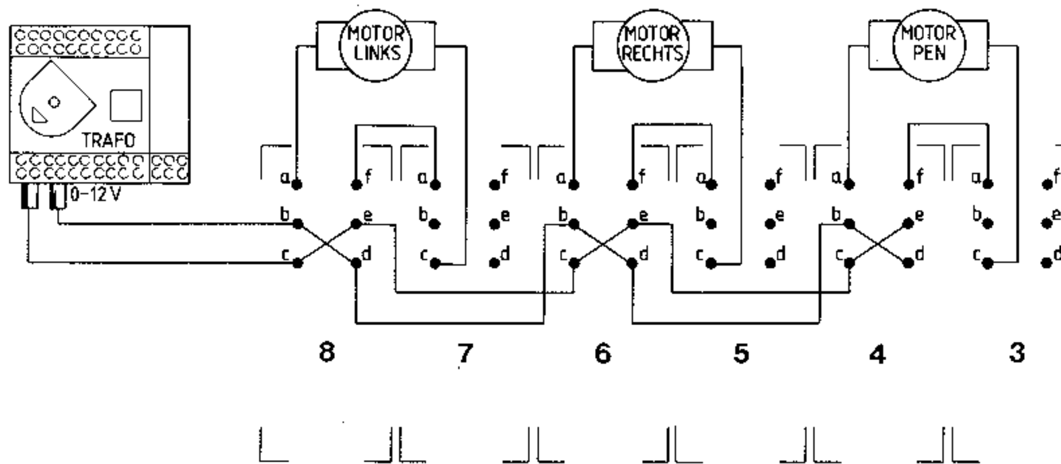
1 = geheugen vullen
2 = film vooruit draaien
3 = film achteruit draaien

Wanneer het programma start, is het geheugen leeg en kan het wagentje nog niets. U kiest uit het menu de optie 1: 'geheugen vullen'. Op het beeldscherm verschijnt het bovenaanzicht van het wagentje. Het is de bedoeling dat de bewegingen van het LEGO-wagentje hetzelfde zijn als die van het wagentje op het scherm. Met behulp van de cursortoetsen bestuurt u het wagentje. Voordat u opdrachten aan het wagentje kunt geven, vraagt het programma eerst of het wagentje direct moet meelopen. De mogelijkheid bestaat om de opdrachten via het beeldscherm te geven, zonder dat het wagentje meerijdt.

Wanneer u klaar bent met uw opdrachten drukt u op F1. Het programma komt dan weer in het menu terecht. Kiest u voor keuze 2 dan gaat het wagentje alle door u opgegeven opdrachten achter elkaar uitvoeren. Ook hier heeft u de keuze om dat eerst via het beeldscherm te doen en daarna het wagentje mee te laten rijden. Het wagentje moet precies dezelfde opdrachten uitvoeren als het wagentje op het scherm. Het is de bedoeling dat bij de opdracht 'vooruit' het wagentje twee wielomwentelingen naar voren gaat. Bij de opdracht 'achteruit' behoort het wagentje twee wielomwentelingen naar achteren te rijden. De tijd waarin een relais 'aan' staat, bepaalt het aantal omwentelingen van de wielen. In de module 'Initialiseren' vindt u de vier variabelen T1, T2, T3 en T4. De waarden van deze variabelen bepalen de tijdsduur dat de motor of motoren ingeschakeld zijn. T1 bepaalt de tijdsduur voor vooruit rijden. T2 bepaalt de tijdsduur voor achteruit rijden. Blijkt het wagentje bij een opdracht 'vooruit' een afstand van vijf wielomwentelingen te maken dan zult u de waarde van variabeleTi moeten verkleinen. Dit zult u proefondervindelijk moeten vaststellen. Het is de bedoeling bij de opdracht 'met de klok mee' of 'tegen de klok in' dat het wagentje precies een hoek van negentig graden maakt. Dit houdt in dat er vier keer een draaiopdracht moet worden gegeven om een volle cirkel te draaien. Door de waarde van de variabelen T3 en T4 te veranderen, kunt u er voor zorgen dat bij de opdrachten steeds het wagentje 90 graden draait.



Bij het overtikken van het programma moet u er op letten dat u de puntkomma achter het LPRINT-statement niet vergeet. Wanneer deze puntkomma ontbreekt, zal het wagentje niet rijden. Met dit wagentje als basis kunt u nog tal van andere experimenten uitvoeren. Aan de hand van de ideeën verderop in het boek kunt u het wagentje uitbouwen tot een complete robot. Met behulp van een derde motor kunt u een viltstift besturen, die u op en neer kunt bewegen. Een foto van zo'n LEGO-mechanisme ziet u in afbeelding 4.3.7. In het programma zult u dan wel een aantal wijzigingen moeten aanbrengen: Het programma moet de opdracht 'pen op' en 'pen neer' accepteren en er zal een module moeten zijn die de opdracht uitvoert. Op de interface zult u de bedrading moeten veranderen zoals in afbeelding 4.3.8.



Afb. 4.3.8 Elektronisch schema van de interface, met veranderde bedrading.

```
10 REM *** BESTURING SCHILDPAD ***
20 CLEAR 1000
30 KEYOFF
40 DIM A(102,2),S(4,2)
50 GOTO 2670
60 :
70 REM *** INITIALISEREN ***
80 DEF USR1=&H156 :REM keybrd leeg
90 DEF USR3=&H41  :REM schrm uit
100 DEF USR4=&H44 :REM schrm aan
110 DEF FNZ(Z) = INT(Z/4)
120 LPRINT CHR$(255)
REM schildpd vooruit REM schildpd achter REM tegen klok in REM met klok
mee REM alles stop REM geheugen leeg
130 VT = 245    : REM schildpd vooruit
140 AC = 240    : REM schildpd achter
150 TK = 241    : REM tegen klok in
160 MK = 244    : REM met klok mee
170 Z  = 255    : REM alles stop
180 N  = 0      : REM geheugen leeg
190 Tl=80 : T2=80 :REM zie boek
200 T3=40 : T4=40 :REM zie boek
210 S(l,l) = 1 : S(l,2) = 2
220 S(2,l) = 5 : S(2,2) = 6
230 S(3,l) = 3 : S(3,2) = 4
240 S(4,l) = 7 : S(4,2) = 8
250 KEY 1,"0" + CHR$(13)
260 E$=CHR$(30)+CHR$(31)+CHR$(28)
270 E$=E$+CHR$(29)+CHR$(24)
280 COLOR 15,4,4 : SCREEN 2,2
290 OPEN "grp:" AS #1
300 GOSUB 400   :REM *** SPRITES INL
310 RETURN
320 :
330 REM *** TEKSTRIJ ***
340 FOR J = 1 TO LEN(A$)
350 C$ = MID$(A$,J,1) i X3=J*6-6+X2
360 DRAW "BM=X3;,=Y2;" : PRINT#1,C$
```

```
370 NEXT J
380 RETURN
390 :
400 REM *** SPRITES INLEZEN ***
410 RESTORE 2780
420 READ SP     :REM aant sprites
430 FOR J = 1 TO SP
440 SD$ = "" : SC = 0
450 FOR I = 1 TO 32
460 READ SD : SC = SC + SD
470 SD$ = SD$ + CHR$(SD)
480 NEXT I
490 READ SD     :REM checksum
500 IF SC = SD THEN 55O
510 SCREEN 0 : COLOR 15,4,4
520 PRINT "Data van sprite";J;
530 PRINT "zijn foutief."
540 STOP
550 SPRITE$(J) = SD$
560 NEXT J
570 RETURN
580 :
590 REM *** VOORUIT ***
600 FOR Q = 1 TO AT
610 IF ML = 1 THEN LPRINT CHR$(VT);
620 E = 0 : RE = FNZ(Tl)
630 IF ML <> 1 THEN ML=RE
640 FOR R = 0 TO Tl STEP ML
650 IF E*RE <> R  OR E>3 THEN 7OO
660 E = E + 1
670 IF F = 1 THEN Y=Y+2*RG
680 IF F = 3 THEN X=X-2*RG
690 GOSUB 1220   :REM *** SPRITES
700 NEXT R
710 IF ML = 1 THEN LPRINT CHR$(Z);
720 NEXT Q
730 RETURN
740 :
750 REM *** ACHTERUIT ***
760 FOR Q = 1 TO AT
770 IF ML = 1 THEN LPRINT CHR$(AC);
780 E = 0 : RE = FNZ(T2)
790 IF ML <> 1 THEN ML=RE
800 FOR R = 0 TO T2 STEP ML
810 IF E*RE <> R OR E>3 THEN 86O
820 E = E + 1
830 IF F = 1 THEN Y=Y-2*RG
840 IF F = 3 THEN X=X+2*RG
850 GOSUB 1220   :REM *** SPRITES
860 NEXT R
870 IF ML = 1 THEN LPRINT CHR$(Z);
880 NEXT Q
890 RETURN
900 :
910 REM *** MET DE KLOK ***
920 FOR Q = 1 TO AT
930 IF ML = 1 THEN LPRINT CHR$(MK);
```

```
940 E = 0 : RE = FNZ(T3)
950 IF ML <> 1 THEN ML=RE
960 FOR R = 0 TO T3 STEP ML
970 IF E*RE <> R OR E>1 THEN 1010
980 E = E + 1 : F = F + 1
990 IF F>4 THEN F=l : RG=RG*-1
1000 GOSUB 1220   :REM *** SPRITES
1010 NEXT R
1020 IF ML = 1 THEN LPRINT CHR$(Z);
1030 NEXT Q
1040 RETURN
1050 :
1060 REM *** TEGEN KLOK IN ***
1070 FOR Q = 1 TO AT
1080 IF ML = 1 THEN LPRINT CHR$(TK)
1090 E = 0 : RE = FNZ(T4)
1100 IF ML <> 1 THEN ML=RE
1110 FOR R = 0 TO T4 STEP ML
1120 IF E*RE <> R OR E>1 THEN 1160
1130 E = E + 1 : F = F - 1
1140 IF F < 1 THEN F=4 : RG=RG*-1
1150 GOSUB 1220   :REM *** SPRITES
1160 NEXT R
1170 IF ML = 1 THEN LPRINT CHR$(Z);
1180 NEXT Q
1190 RETURN
1200 :
1210 REM *** SPRITES ***
1220 IF Y = 208 THEN 1270
1230 IF X > 255  THEN X=10
1240 IF X < 10 THEN  X=255
1250 IF Y > 150 THEN Y=0
1260 IF Y < 0   THEN Y=150
1270 PUT SPRITE 1,(X,Y),10,S(F,1)
1280 PUT SPRITE 2,(X,Y),1,S(F,2)
1290 RETURN
1300 :
1310 REM *** BEELDSCHERM ***
1320 COLOR 1 : CLS
1330 LINE (0,160)-(255,160),1
1340 LINE (0,0)-(255,0),1
1350 LINE (0,161)-(255,190),10,BF
1360 X2=20 : Y2=164 : GOSUB 340
1370 RETURN
1380 :
1390 REM *** VRAAG ***
1400 LINE (40,175)-(255,183),10,BF
1410 A$ = B$ + " <J/N>"
1420 L = LEN(A$)*6+45
1430 X2 = 40 : Y2 = 175 : GOSUB 340
1440 CR = 10 : TIME = 0
1450 DUMMY = USR1(0)
1460 IF TIME MOD 15 <> 0 THEN 150O
1470 IF CR = 10 THEN CR=1 ELSE CR=10
1480 COLOR CR
1490 DRAW"BM=L;,175" : PRINT#1,"?"
1500 A$=INKEY$ : IF A$="" THEN 146O
```

```
1510 RS = INSTR("JjNn",A$)
1520 IF RS = 0 THEN 1460
1530 IF RS > 2 THEN RS=2 ELSE RS=1
1540 LINE (40,175)-(255,183),10,BF
1550 COLOR 1
1560 RETURN
1570 :
1580 REM *** FILM VOORUIT ***
1590 IF N = 0 THEN 1780
1600 A$ = "-Film vooruit-"
1610 GOSUB 1320 :REM *** BEELDSCH
1620 B$="Moet schildpad meelopen"
1630 GOSUB 1400 :REM *** VRAAG
1640 IF RS=1 THEN ML=1 ELSE ML=50
1650 X = 125 : Y = 95 I F=l : RG=-1
1660 GOSUB 1220  :REM *** SPRITES
1670 FOR I = 1 TO N
1680 AT = A(I,2) :REM onthouden
1690 IF A(I,1) = 1 THEN GOSUB 600
1700 IF A(I,1) = 2 THEN GOSUB 760
1710 IF A(I,1) = 3 THEN GOSUB 920
1720 IF A(I,1) * 4 THEN GOSUB 1070
1730 NEXT I
1740 B$ = "Nog een keer"
1750 GOSUB 1400 :REM *** VRAAG
1760 IF RS = 1 THEN 1600
1770 Y = 208 : GOSUB 1220
1780 RETURN
1790 :
1800 REM *** FILM ACHTERUIT ***
1810 IF N = 0 THEN 2010
1820 A$ = "-Film achteruit-"
1830 GOSUB 1320 :REM *** BEELDSCH
1840 B$="Moet schildpad meelopen"
1850 GOSUB 1400 :REM *** VRAAG
1860 IF RS=1 THEN ML=1 ELSE ML=50
1870 X = A(101,1) : Y = A(101,2)
1880 F = A(102,l) : RG = A(102,2)
1890 GOSUB 1220   :REM *** SPRITES
1900 FOR I = N TO 1 STEP -1
1910 AT = A(I,2) :REM onthouden
1920 IF A(I,1) = 1 THEN GOSUB 760
1930 IF A(I,1) = 2 THEN GOSUB 6OO
1940 IF A(I,1) = 3 THEN GOSUB 107O
1950 IF A(I,1) = 4 THEN GOSUB 920
1960 NEXT I
1970 B$ = "Nog een keer"
1980 GOSUB 1400 :REM *** VRAAG
1990 IF RS = 1 THEN 182O
2000 Y = 208 : GOSUB 1220
2O1O RETURN
2020 :
2030 REM *** GEHEUGEN VULLEN ***
2040 FOR I = 1 TO 100
2050 A(I,1)=0 : A(I,2)=0 :REM wissen
2060 NEXT I
2O70 N = 0   : AT = 1  : RG = -1
```

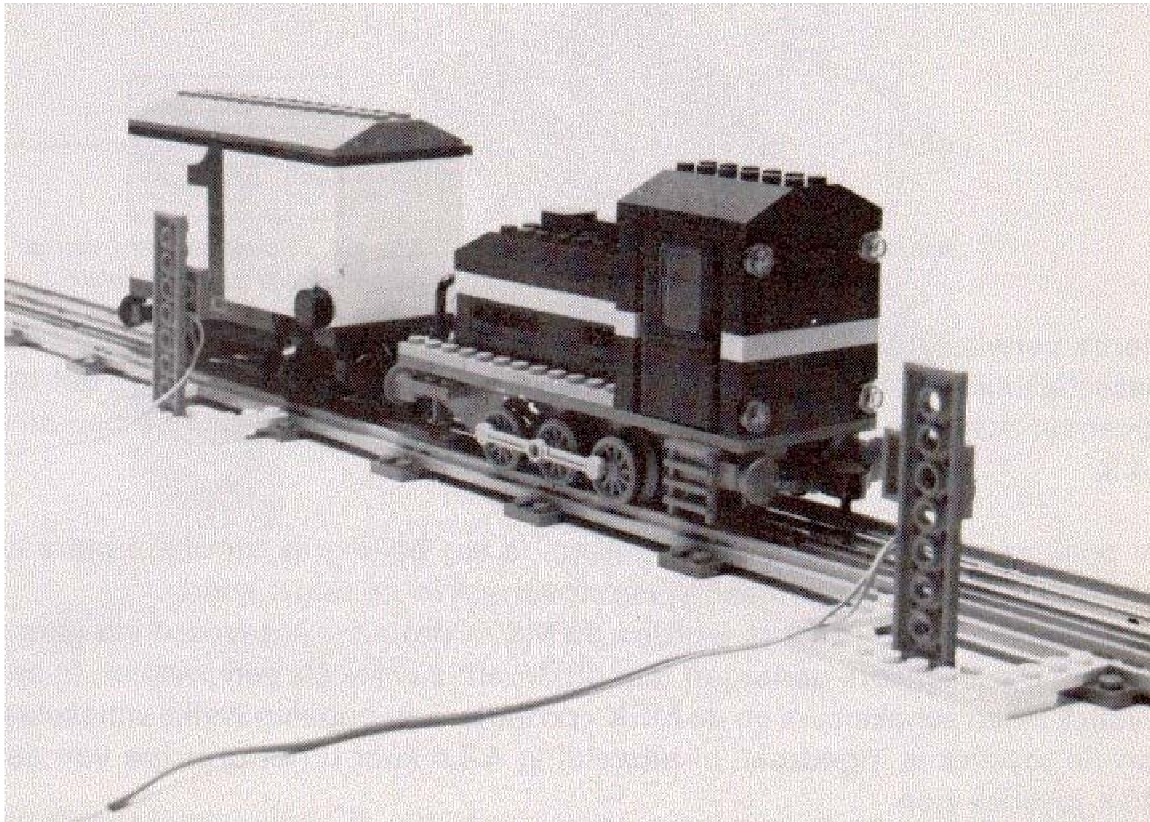
```
2080 X = 125 : Y = 95 : F = 1
2090 A$ = "-Geheugen vullen-"
2100 GOSUB 1320
2110 GOSUB 1220  :REM *** SPRITES
2120 B$="Moet schildpad meelopen"
2130 GOSUB 1400
2140 IF RS=1 THEN ML=1 ELSE ML=5O
2150 A$ = "Opdracht :" : X2 = 14O
2160 Y2 = 170 : GOSUB 340
2170 CR = 1 : TIME = 0
2180 IF TIME MOD 25 <> 0 THEN 2220
2190 IF CR = 1 THEN CR=10 ELSE CR=1
2200 COLOR CR : DRAW "BM200,170"
2210 PRINT #1,"*"
2220 A$=INKEY$ : IF A$="" THEN 2180
2230 REM ** WELK COMMANDO **
2240 P = INSTR(E$,A$)
2250 IF P = 0 THEN 2180
2260 IF P = 5 THEN 2330
2270 ON P GOSUB 600,760,920,1070
2280 IF N = 0 THEN 2300
2290 IF P = A(N,1) THEN 2310
2300 N = N + 1 : A(N,1) = P
2310 A(N,2) = A(N,2) + 1
2320 IF N < 100 THEN 2170
2330 A(101,1) = X : A(101,2) = Y
2340 A(102,1) = F : A(102,2)  = RG
2350 Y = 208 : GOSUB 1220 : COLOR 1
2360 RETURN
2370 :
2380 REM *** MENU ***
2390 CLS : COLOR 1
2400 A$ = "- Menu -" : X2 = 20
2410 Y2 - 10 : GOSUB 340
2420 A$ = "0 = Stoppen  of [Fl]"
2430 X2 = 50 : Y2 = 40 : GOSUB 340
2440 A$ = "1 = Geheugen vullen"
2450 Y2 = 60 : GOSUB 340
2460 A$ = "2 = Film vooruit "
2470 Y2 = 80 : GOSUB 340
2480 A$ = "3 = Film achteruit"
2490 Y2 = 100 : GOSUB 340
2500 A$ = "Wat is uw keuze "
2510 X2 = 20 : Y2 = 140 : GOSUB 340
2520 DUMMY = USR1(0) : CR = 1
2530 TIME = 0
2540 IF TIME MOD 15 <> 0 THEN 2580
2550 IF CR = 1 THEN CR=0 ELSE CR=1
2560 COLOR CR : DRAW"BM115,140"
2570 PRINT #1,"?"
2580 A$=INKEY$ : IF A$="" THEN 2540
2590 M=INSTR("0123",A$) - 1
2600 IF M = -1 THEN 2540
2610 LINE (115,140)-(123,150),4,BF
2620 COLOR 1 : DRAW "BM115,140"
2630 PRINT#1,A$
2640 RETURN
```

```
2650 :
2660 REM **** STUURMODULE ****
2670 GOSUB 80   :REM *** INT
2680 GOSUB 2390 :REM *** MENU
2690 IF M = 1 THEN GOSUB 2040
2700 IF M = 2 THEN GOSUB 1590
2710 IF M = 3 THEN GOSUB 1810
2720 IF M <> 0 THEN 2680
2730 COLOR 15,4,4 : CLS
2740 END
2750 :
2760 :
2770 REM @@@ INTERNE DATAFILE @@@
2780 DATA 8     :REM aantal sprites
2790 :
2800 REM @@@ SPRITE 1 @@@
2810 DATA 0,0,0,31,31,31
2820 DATA 31,63,63,31,31,31
2830 DATA 31,0,0,0,0,0
2840 DATA 0,248,248,248,248,252
2850 DATA 252,248,248,248,248,0
2860 DATA 0,0
2870 DATA 2862  :REM checksum
2880 :
2890 REM @@@ SPRITE 2 @@@
2900 DATA 0,0,0,0,0,192
2910 DATA 192,192,192,192,192,0
2920 DATA 0,0,0,0,0,0
2930 DATA 0,0,0,3,3,3
2940 DATA 3,3,3,0,0,0
2950 DATA 0,0
2960 DATA 1170  :REM checksum
2970 :
2980 REM @@@ SPRITE 3 @@@
2990 DATA 0,0,1,31,31,31
3000 DATA 31,31,31,31,31,31
3010 DATA 31,1,0,0,0,0
3020 DATA 128,248,248,248,248,248
3030 DATA 248,248,248,248,248,128
3040 DATA 0,0
3050 DATA 3048  :REM checksum
3060 :
3070 REM @@@ SPRITE 4 @@@
3O80 DATA 7,7,0,0,0,0
3O90 DATA 0,0,0,0,0,0
3100 DATA 0,0,7,7,224,224
3110 DATA 0,0,0,0,0,0
3120 DATA 0,0,0,0,0,0
3130 DATA 224,224
3140 DATA 924   :REM checksum
3150 :
3160 REM @@@ SPRITE 5 @@@
3170 DATA 0,0,1,15,15,15
3180 DATA 31,63,31,15,7,3
3190 DATA 1,0,0,0,0,128
3200 DATA 192,224,240,248,252,254
3210 DATA 252,248,248,248,192,128
```

```
3220 DATA 0,0
3230 DATA 3O51  :REM checksum
3240 :
3250 REM @@@ SPRITE 6 @@@
3260 DATA 8,28,56,112,224,64
3270 DATA 0,0,0,0,0,0
3280 DATA 0,0,0,0,0,0
3290 DATA 0,0,0,0,0,0
3300 DATA 0,0,2,7,14,28
3310 DATA 56,16
3320 DATA 615   :REM checksum
3330 :
3340 REM @@@ SPRITE 7 @@@
3350 DATA 0,1,3,7,15,31
3360 DATA 63,127,63,31,31,31
3370 DATA 3,1,0,0,0,0
3380 DATA 128,240,240,240,248,252
3390 DATA 248,240,224,192,128,0
3400 DATA 0,0
3410 DATA 2787  :REM checksum
3420:
3430 REM @@@ SPRITE 8 @@@
3440 DATA 0,0,0,0,0,0
3450 DATA 0,0,0,0,64,224
3460 DATA 112,56,28,8,16,56
3470 DATA 28,14,7,2,0,0
3480 DATA 0,0,0,0,0,0
3490 DATA 0,0
3500 DATA 615   :REM checksum
```

4.4 Spoorbaan

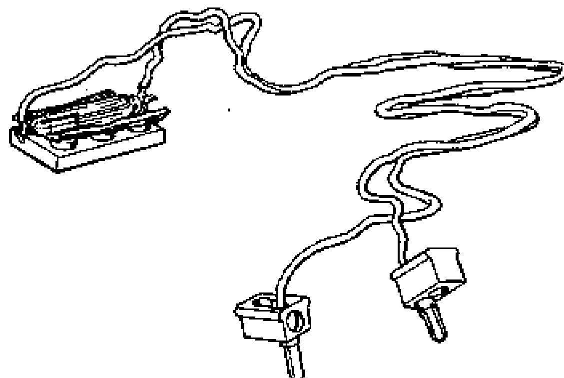
In het programma 'Schildpad' worden signalen van de computer naar het wagentje gestuurd. Er komen geen signalen terug van het wagentje. Er is eigenlijk sprake van sturing en niet van besturing. In het programma 'Spoorbaan' komt besturing aan de orde. De MSX-computer regelt de rijrichting van de trein, en sensors langs de treinbaan geven signalen af als de trein er langs komt. Met behulp van de signalen van de sensors wordt de trein bestuurd.



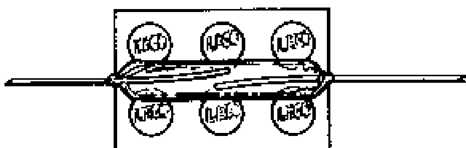
Afb. 4.4.1. De locatie van de positie sensors langs de baan. Let ook op het ronde magneetje dat op het wagentjes is gemonteerd.

Op een afstand van zo'n veertig tot zestig centimeter uit elkaar zijn twee sensors geplaatst. Op het moment dat er een trein langs rijdt, geeft de sensor een signaal af. Dit signaal wordt door de MSX-computer opgevangen en in het programma verwerkt. De computer zet de trein een aantal seconden stil en daarna rijdt de trein in de tegenovergestelde richting weg. Het doel van het programma is om op eenvoudige wijze de mogelijkheden van besturing op een MSX-computer uiteen te zetten.
De sensors zijn opgebouwd uit een aantal onderdelen. U heeft hiervoor nodig: dubbelzijdig plakband, twee glasschakelaars, vier stukken soepel montagedraad, vier stekkertjes en twee platte zesnops LEGO-steentjes. Het dubbelzijdig plakband is het soort plakband waarmee vloerbedekking op de grond wordt vastgeplakt. De

glasschakelaars kunt u in elke elektronicazaak kopen. Deze schakelaars zijn twee centimeter lang en hebben een doorsnede van 0,5 centimeter.
Aan de hand van afbeelding 4.4.2 kunt u zien hoe u een glasschakelaar op een plat zesnops steenje bevestigt. U knipt daartoe van het dubbelzijdig plakband een strook van 2 bij 0,5 centimeter af. Het strookje plakband drukt u stevig tussen de noppen van het LEGO-steentje, zoals op de tekening is aangegeven.



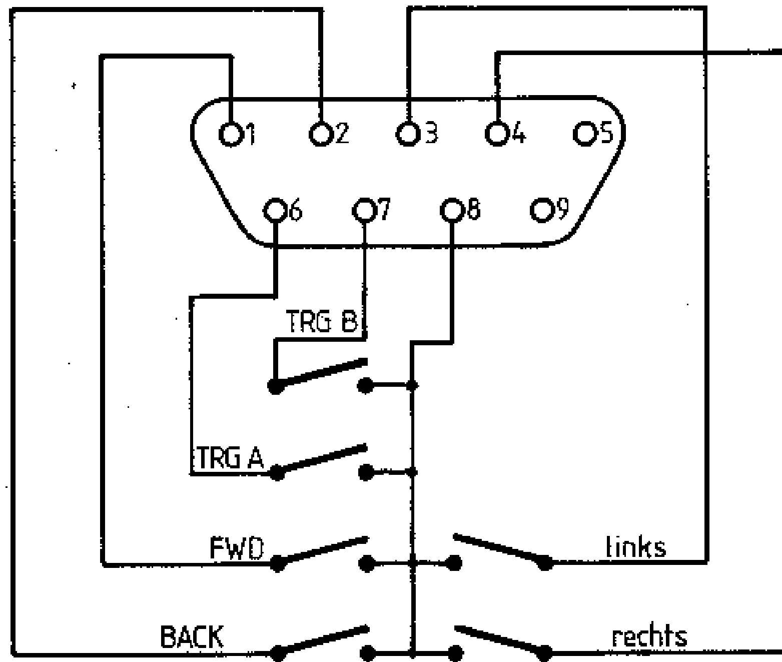
Afb. 4.4.2



Afb. 4.4.3 Bovenaanzicht van de gemonteerde glasschakelaar.

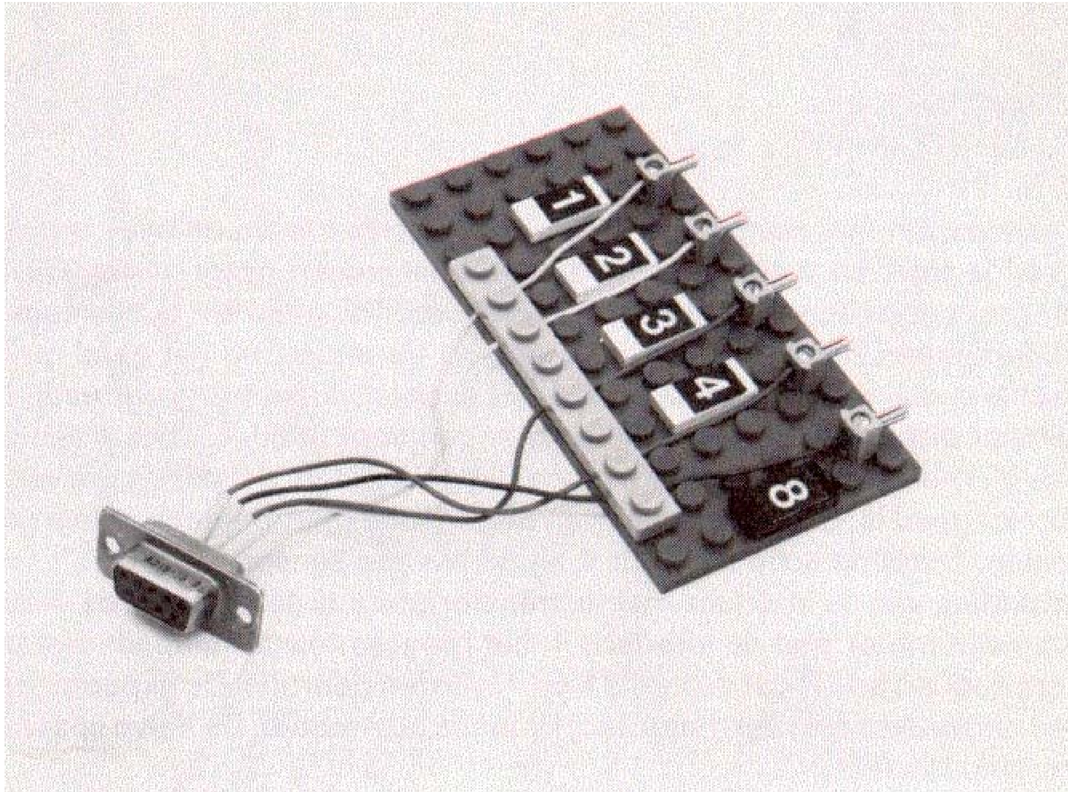
Hierna verwijdert u de beschermlaag, zodat de andere plaklaag vrijkomt. De glasschakelaar wordt hierop stevig vastgedrukt. U moet er op letten dat het bovenaanzicht van het steentje met schakelaar hetzelfde is als in afbeelding 4.4.3.

De positionering van de glasschakelaars is erg belangrijk, omdat anders de schakelaar niet sluit, wanneer er een magneet in de buurt komt. De signalen van de sensors worden de MSX-computer ingevoerd via gamepoort 1. Deze sensors gedragen zich op dezelfde wijze als een joystick. In een joystick zitten schakelaars en de MSX-computer kan bepalen welke schakelaar van de joystick is ingedrukt. In afbeelding 4.4.4 kunt u het principe van een MSX-joystick zien.



Afb. 4.4.4 Schema van een MSX-joystick.

Wanneer een magneet langs een glasschakelaar wordt gehaald, sluit deze schakelaar. Van dit gegeven wordt in dit programma gebruik gemaakt. Eerst zult u ervoor moeten zorgen dat u de signalen in de MSX-computer kunt op vangen. Hiertoe heeft u een negenpolige D-connector (een vrouwtje) nodig.

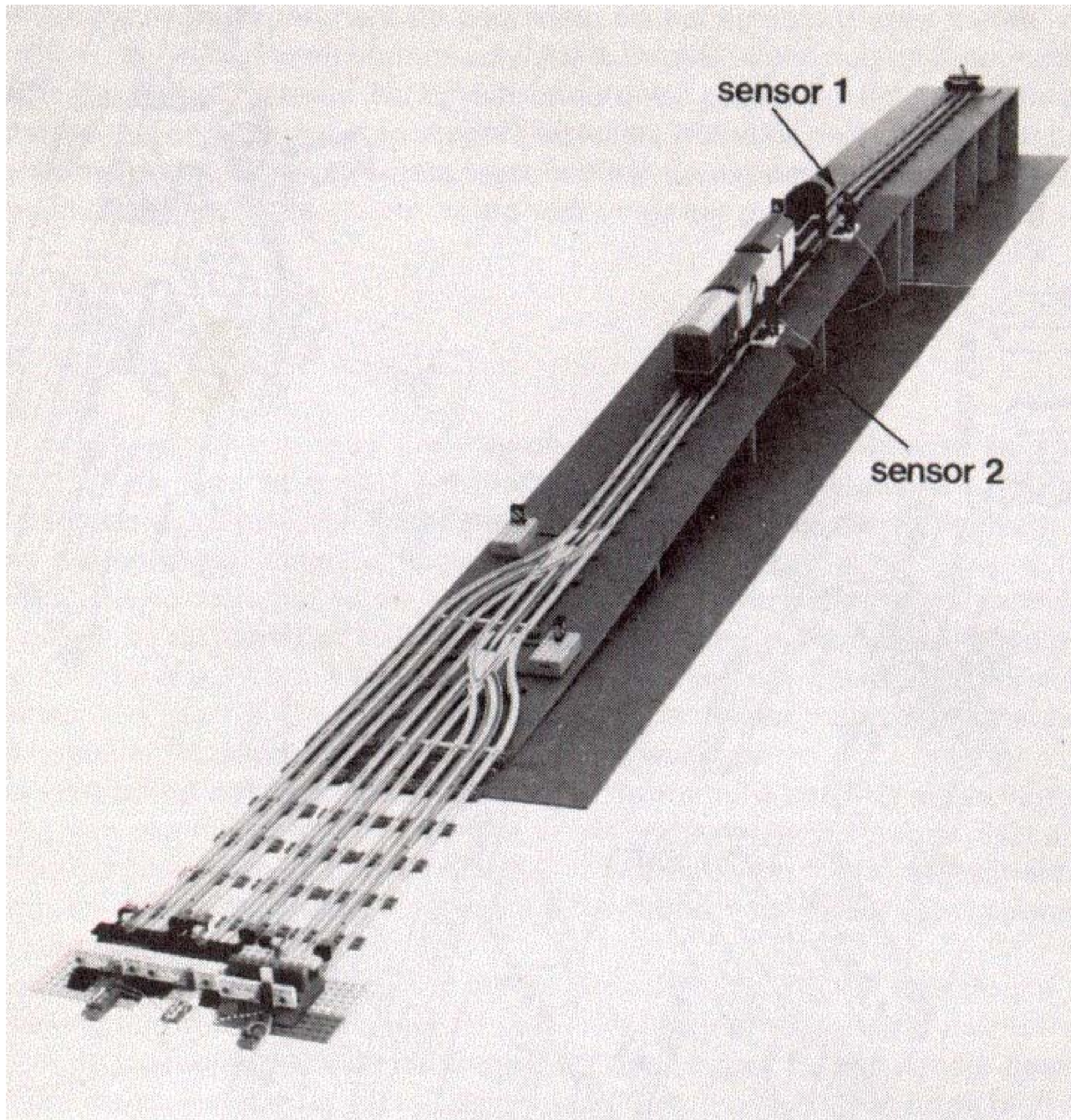


Aan de achterzijde van de connector staan nummers, deze nummers komen overeen met de nummers in afbeelding 4.4.4. U knipt nu vijf draadjes van ongeveer 15 centimeter en soldeert deze op de punten 8,4,3,2 en 1 van de connector. Aan de andere kant bevestigt u stekkertjes om een gemakkelijke verbinding tot stand te brengen. Met behulp van etiketjes of iets dergelijks geeft u per draadje aan op welk pennetje van de D-connector deze is gesoldeerd. In afbeelding 4.4.5 is te zien hoe u dit bijvoorbeeld kunt doen. Als laatste moet u een rond LEGO-magneetje met dubbelzijdig plakband op de zijkant van een normaal viernops steentje plakken. De ronde LEGO-magneetjes worden gebruikt voor de koppelingen van de wagons en passen precies op een viernops steentje. Bij dit treinexperiment is uitgegaan van een 12 volts LEGO-trein. Bij het bouwen van het wagonnetje moet het LEGO-steentje met het ronde magneetje direct op de grondplaat zitten aan de rechterzijde, zie afbeelding 4.4.1. Voor de proefbaan heeft u een recht stuk baan nodig van zo'n centimeter of zestig. Langs de baan plaatst u twee sensoren. De afstand tussen deze twee sensoren is 40 centimeter. De foto van de treinbaan verduidelijkt dit. Hierna verbindt u een draad van sensor 1 met stekkertje nummer 3. De andere draad van dezelfde sensor gaat naar stekker 8. E,n draad van sensor 2 gaat naar stekker 1. De tweede draad van sensor 2 gaat naar stekker 8. Op stekker nummer 8 zitten nu twee stekkers, ,,n van sensor 1 en ,,n van sensor 2.

U kunt voor dit experiment de bedrading op het speldenkussen handhaven zoals u dat gedaan heeft voor de schildpad. U moet alleen nog de D-connector in gamepoort nr. 1 steken. Nu zijn alIe onderdelen gereed om het experiment te starten. Als eerste moet u de sensors instellen. Hiervoor heeft u een klein programma nodig. Dit programma leest continu gamepoort 1 uit en plaatst het resultaat op het beeldscherm. Hieronder volgt het programma:

```
10 CLS
20 LOCATE 10,10
30 PRINT STICK(l);" "
40 GOTO 20
```

U zet een wagon met magneetje op de rails precies tussen de twee sensors, zoals op de foto, en u start het hierboven beschreven programma. Op het beeldscherm verschijnen achtereenvolgens dezelfde getallen. U beweegt nu het wagentje met magneet langs een sensor. Verandert nu het getal op het beeldscherm dan staat die sensor goed. Als er geen verandering optreedt, zult u de sensor dichterbij de baan moeten brengen door er bijvoorbeeld een plat steentje bij te plaatsen. Een andere mogelijkheid kan zijn, dat de hoogte van de sensor niet goed is. U zult de plaats van de sensor experimenteel moeten zien te bepalen. Dezelfde actie herhaalt u bij de tweede sensor. De verbinding van de interface met de transformator is dezelfde als de verbinding van een motor met de interface in het programma 'Schildpad'. Het programma gaat ervan uit dat de relais 8 en 7 worden gebruikt. De experimenten worden uitgevoerd met een langzaam rijdende locomotief. De halve spanning is ruim voldoende.
Zoals reeds eerder is gezegd, is het de bedoeling om met dit experiment de werking van sensoren te demonstreren. Het programma zorgt ervoor dat de trein de tegenovergestelde richting wordt opgestuurd als een sensor geactiveerd wordt. Het gevolg zal zijn dat de trein tussen de twee sensoren blijft rijden. Dit basisidee kunt u gebruiken om een kleinetreinbaan met behulp van de computer aan te sturen. In dit boek zijn de idee‰n uitgewerkt met een LEGO-trein, maar het moet ook mogelijk zijn hetzelfde te bereiken met een normale HO-trein.
Het programma spreekt eigenlijk voor zich. Voor het experiment is een locomotief nodig en een wagon met magneetje. Op het beeldscherm krijgt u deze locomotief en wagon schematisch te zien evenals de sensoren. Op het beeldscherm verschijnt de vraag of de trein tussen de sensoren staat. U moet ervoor zorgen dat de trein daar inderdaad staat en daarna geeft u Ja. De trein zal een bepaalde richting oprijden totdat een van twee sensors de trein detecteert. De MSX-computer geeft dan de opdracht 'stoppen'. Na deze detectie kunt u opgeven hoe vaak u de trein op en neer wilt laten rijden. U geeft een cijfer tussen 1 en 9. Hierna wordt de opdracht uitgevoerd. De modulaire opbouw en de lengte van het programma maken het mogelijk om met een klein beetje inspanning de werking van het programma te doorgronden, zodat u zelf ook dit soort programma's kunt schrijven.
Suggestie: Het is mogelijk om met de modules uit dit programma een compleet werkende rangeerheuvel te bouwen. Om u een stuk op weg te helpen,hebben we een foto en tekening afgebeeld.




```
10 REM *** PROEFBAAN ***
20 DIM A(2,3)
30 KEYOFF
40 GOTO 1530
50 :
60 REM *** INITIALiSEREN ***
70 DEF USR1=&H156 :REM keybrd leeg
80 DEF USR3=&H41  :REM schrm uit
90 DEF USR4=&H44  :REM schrm aan
100 A(1,1) = 252  :REM richting l
110 A(2,1) = 253  :REM richting 2
120 A(1,2) = 160  :REM spriteplts
130 A(2,2) = 60   :REM spritaplts
140 A(1,3) = 1    :REM sensor l
150 A(2,3) = 5    :REM sensor 5
160 KEY 1,"?"
170 COLOR 15,4,4 : SCREEN 2,2
```

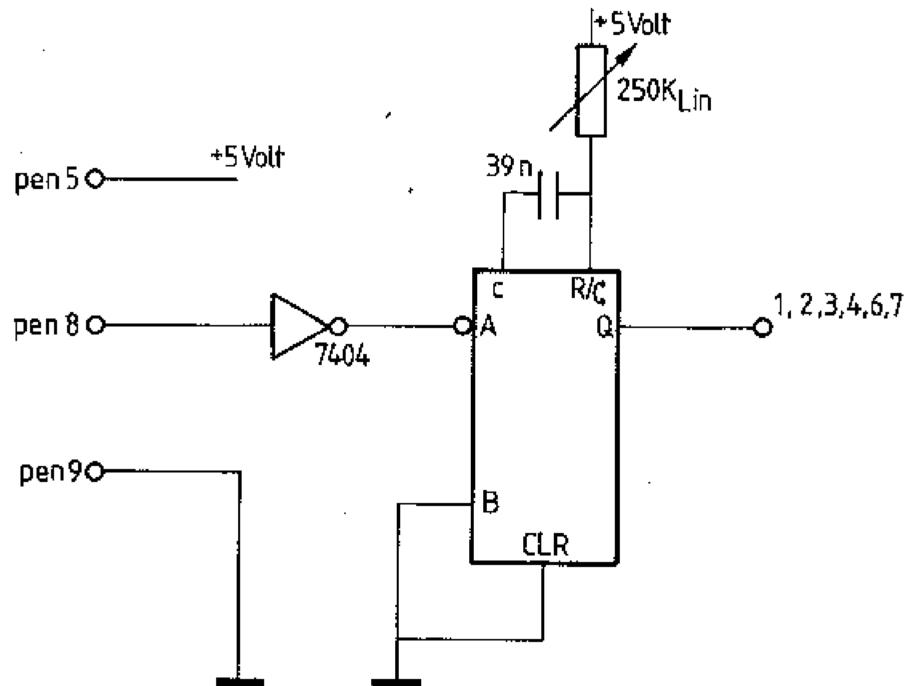
```
180 OPEN "grp:" AS #1
190 GOSUB 290   :REM *** SPRITES INL
200 RETURN
210 :
220 REM *** TEKSTRIJ ***
230 FOR J = 1 TO LEN(A$)
240 C$ = MID$(A$,J,1)  : X3=J*6-6+X2
250 DRAW "BM=X3;,=Y2;" : PRINT#1,C$
260 NEXT J
270 RETURN
280 :
290 REM *** SPRITES INLEZEN ***
300 RESTORE 1630
310 READ SP  :REM aant sprites
320 FOR J = 1 TO SP
330 SD$ = "" : SC = 0
340 FOR I = 1 TO 32
350 READ SD  : SC = SC + SD
360 SD$ = SD$ + CHR$(SD)
370 NEXT I
380 READ SD     :REM checksum
390 IF SC = SD THEN 440
400 SCREEN 0 : COLOR 15,4,4
410 PRINT "Data van sprite";J;
420 PRINT "zijn foutief."
430 STOP
440 SPRITE$(J) = SD$
450 NEXT J
460 RETURN
470 :
480 REM *** SPRITES ***
490 PUT SPRITE 1,(X,Y),1,2
500 PUT SPRITE 2,(X+18,Y),1,1
510 RETURN
520 :
530 REM *** BEELDSCHERM ***
540 DUMMY = USR3(0) :REM schrm uit
550 COLOR 1 : CLS
560 LINE (0,160)-(255,160),1
570 LINE (0,161)-(255,190),10,BF
580 A$ = "Proefbaan"
590 X2=20 : Y2=164 : GOSUB 230
600 X = 100 : Y = 83 : GOSUB 490
610 LINE(40,100)-(200,100),1
620 LINE(70,70)-(70,110),10
630 LINE(170,70)-(170,110),10
640 A$ = "SR-1" : X2 = 58
650 Y2 = 60 : GOSUB 230
660 A$ = "SR-2" : X2 = 158
670 Y2 = 60 : GOSUB 230
680 DUMMY = USR4(0) :REM schrm aan
690 RETURN
700 :
710 REM *** HOE VAAK ***
720 LINE(30,175)-(255,183),10,BF
730 A$="Hoe vaak op en neer F1=stop"
740 X2 = 30 : Y2 = 175 : GOSUB 230
```

```
750 CR=10 : TIME=0 : DUMMY=USR1(0)
760 IF TIME MOD 15 <> 0 THEN 800
770 IF CR=10 THEN CR=1 ELSE CR=10
780 COLOR CR
790 DRAW "BM212,175" : PRINT#1,"?"
800 A$=INKEY$ : IF A$=""THEN 760
810 AT=INSTR("?1234567890",A$)-1
820 IF AT = -1 THEN 760
830 LINE(212,175)-(220,185),10,BF
840 DRAW "BM212,175":PRINT#1,A$
850 RETURN
860 :
870 REM *** JA/NEE ***
880 LINE (40,175)-(255,183),10,BF
890 A$ = B$ + " <J/N>"
900 L = LEN(A$)*6+45
910 X2 = 40 : Y2 = 175 : GOSUB 230
920 CR = 10 : TIME = 0
930 DUMMY = USR1(0) :REM keybrd leeg
940 IF TIME MOD 15 <> 0 THEN 980
950 IF CR = 10 THEN CR=1 ELSE CR=10
960 COLOR CR
970 DRAW"BM=L;,175" : PRINT#1,"?"
980 A$=INKEY$ : IF A$="" THEN 940
990 RS = INSTR("JjNn",A$)
1000 IF RS = 0 THEN 940
1010 IF RS > 2 THEN RS=2 ELSE RS=1
1020 LINE (40,175)-(255,183),10,BF
1030 COLOR 1
1040 RETURN
1050 :
1060 REM *** OUTPUT ***
1070 LPRINT CHR$(PT); :REM let op ;
1080 RETURN
1090 :
1100 REM *** SENSOR ***
1110 ST = STICK(L)
1120 IF ST=0 OR ST=SR THEN 1110
1130 RETURN
1140 :
1150 REM *** START ***
1160 B$="Staat trein tussen sensors"
1170 GOSUB 880   :REM *** VRAAG
1180 IF RS = 2 THEN 1160
1190 H=1 : PT = A(H,1)
1200 SR = 0
1210 GOSUB 1070   :REM *** OUTPUT
1220 GOSUB 1110   :REM *** SENSOR
1230 PT = 255     :REM trein stoppen
1240 GOSUB 1070   :REM *** OUTPUT
1250 IF ST=5 THEN 1270
1260 H = 2 : SWAP A(1,1),A(2,1)
1270 X = A(H,2) : GOSUB 490
1280 RETURN
1290 :
1300 REM *** BAANBESTURING ***
1310 COLOR 1 : GOSUB 720
```

```
1320 IF AT = 0 THEN 1500
1330 COLOR 1
1340 LINE(30,175)-(255,183),10,BF
1350 A$="Baanbesturing" : X2 =30
1360 Y2 = 175 : GOSUB 230
1370 FOR I = 1 TO AT
1380 IF H=1 THEN H=2 ELSE H=1
1390 SR = A(H,3)
1400 PT = A(H,1) : X = A(H,2)
1410 GOSUB 1070 :REM *** OUTPUT
1420 GOSUB 1110 :REM *** SENSOR
1430 GOSUB 490  :REM *** SPRITES
1440 PT = 255
1450 GOSUB 1070 :REM *** OUTPUT
1460 TIME = 0 : T2 = 2 :REM wachtd
1470 T3 = T2*50 - TIME
1480 IF T3 > 0 THEN 1470
1490 NEXT I
1500 RETURN
1510 :
1520 REM **** STUURMODULE ****
1530 GOSUB 70   :REM *** INIT
1540 GOSUB 540  :REM *** BEELDSCH
1550 GOSUB 1160 :REM *** START
1560 GOSUB 1310 :REM *** BAANBESTUR
1570 IF AT <> 0 THEN 1560
1580 COLOR 15,4,4 : CLS
1590 END
1600 :
1610 :
1620 REM @@@ INTERNE DATAFILE @@@
1630 DATA 2     :REM aantal sprites
1640 :
1650 REM @@@ SPRITE l @@@
1660 DATA 0,0,0,0,0,99
1670 DATA 255,255,255,255,0,255
1680 DATA 16,41,16,0,0,0
1690 DATA 31,17,17,17,255,255
1700 DATA 255,255,0,255,132,74
1710 DATA 132,0
1720 DATA 3142  :REM checksum
1730 :
1740 REM @@@ SPRITE 2 @@@
1750 DATA 0,0,255,127,127,127
1760 DATA 127,127,127,127,127,127
1770 DATA 32,80,32,0,0,0
1780 DATA 255,254,254,254,254,254
1790 DATA 254,254,254,254,4,10
1800 DATA 4,0
1810 DATA 4101  :REM checksum
```

4.5 Interface 2

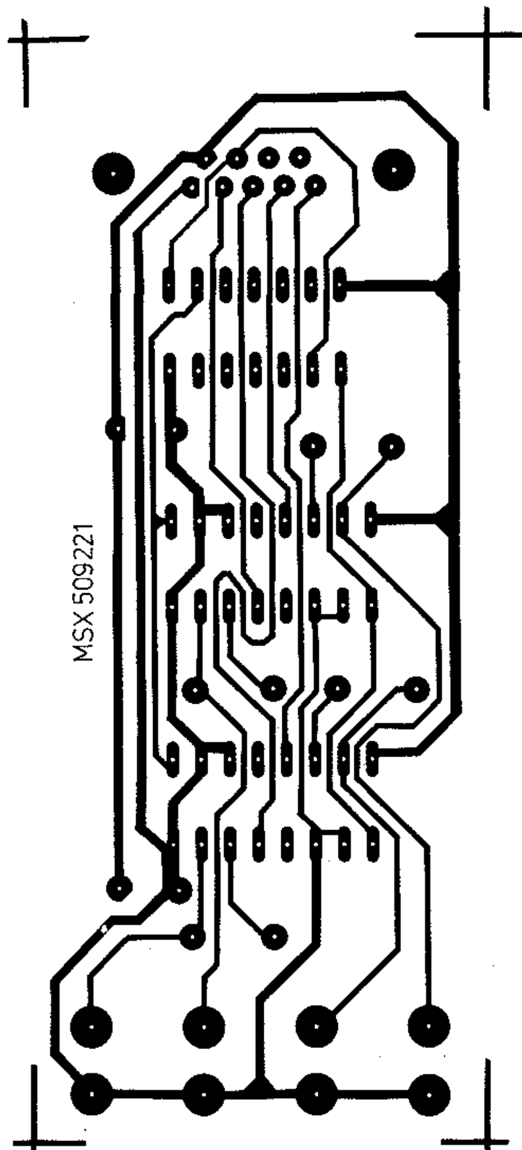
Wanneer u de documentatie van uw MSX-computer over de gamepoort naleest, moet het u zijn opgevallen dat een aantal krachtige BASIC-statements deze poort ondersteunt. In de eerste plaats is dat het bekende STICK-state-ment. Behalve het genoemde statement is er ook nog een PDL-functie en een PAD-functie. Deze twee zijn minder bekend. Van deze twee laatste functies is de PDL-functie het interessantst. Deze functie verzorgt de uitlezing van een zgn. 'game-paddle'.
Een game-paddle is eigenlijk een variabele weerstand in een fraaie behuizing. Wordt de weerstand verdraaid, dan verandert direct de waarde van de PAD-functie. De rest van de informatie, hoe dit geschiedt en welke maximale weerstandswaarde kan worden uitgelezen, ontbrak.
In een publicatie van Microsoft, de ontwerpers van het MSX-concept, werd echter uiteengezet hoe de game-paddle werkt. In het kort komt het erop neer dat de MSX-computer naar pin 8 van de game-poort een puls stuurt. Deze puls activeert een signaal in een externe elektronische schakeling. Het signaal uit deze schakeling wordt opnieuw gedecteerd door de MSX-computer. De tijdsduur van dit signaal bepaalt de waarde die de PDL-functie krijgt. De genoemde tijdsduur is een functie van de weerstandswaarde van de variabele weerstand. In afbeelding 4.5.1 vindt u het principeschema van de schakeling. Tevens is een kleine printkaart met componentenopdruk afgedrukt (afbeeldingen 4.5.2 en 4.5.3), waarop vier game-paddles kunnen worden aangesloten. Met behulp van twee van dit soort printen kan de MSX-computer acht paddles ondersteunen.



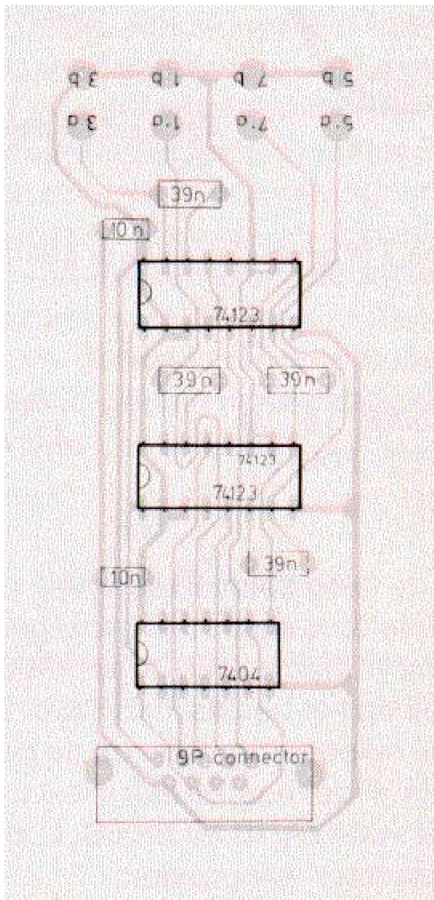
Afb. 4.5.1 Het elektronisch schema.

Om de schakeling te maken, heeft u de volgende onderdelen nodig:

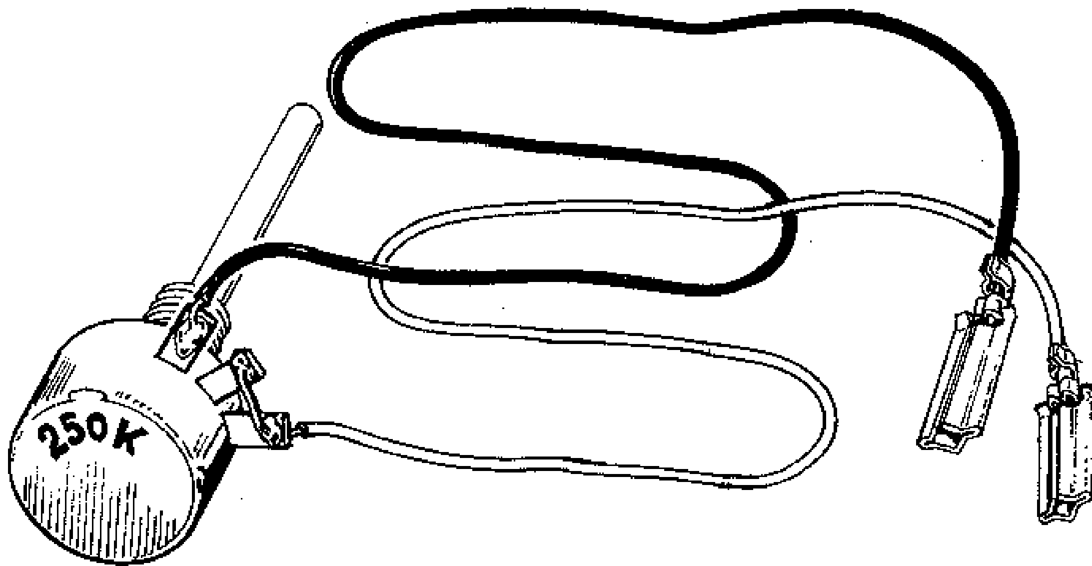
- 1 9-polige D-connector haaks (vrouwtje).
- 2 74123 of 74LS123.
- 1 7404 of 74LS04.
- 4 MKM-condensator 39 nF.
- 3 10 nF ontkoppelcondensator.
- 8 soldeerpennen.
- 2 250 kilo-ohm lineaire draaipotentiometers.



Afb. 4.5.2 De print-layout



Aan de hand van de componenten-layout is het vrij eenvoudig deze schakeling in elkaar te zetten. Hiertoe soldeert u eerst de 9-polige D-connector op de print-kaart en de acht soldeerpennen. Hierna komen de vier condensatoren van 39 nF en de drie ontkoppelcondensatoren. Ten slotte kunt u de drie IC's op de print solderen. Het is verstandiger om printvoetjes te solderen en hierin de IC's te drukken. U moet erop letten dat de IC's in de goede richting worden geplaatst.
Aan de potentiometers soldeert u twee stukken soepel montagedraad van zo'n 30 cm lengte en twee soldeerklemmen. In afbeelding 4.5.4 is aangegeven hoe u de draden aan de potentiometer moet solderen.



Afb. 4.5.4 De potentiometer met soldeerklemmen.

Interface 2 is gereed en u kunt gaan testen. Hieronder vindt u het testprogramma.

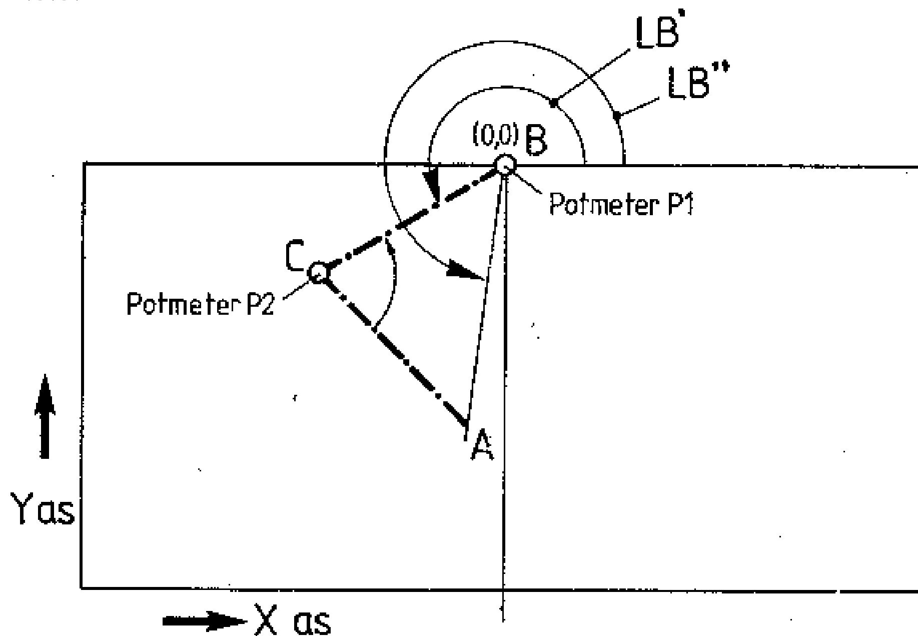
```
10 CLS
20 LOCATE 10,10
30 PRINT PDL(3);" "
40 GOTO 20
```

U prikt de twee soldeerklemmen van een potentiometer op de pennen 3A en 3B. De interface steekt u in gamepoort 1 van de MSX-computer. Wanneer u het programma start en aan de potentiometer gaat draaien, moet u de getallen van O t/m 255 op het beeldscherm zien verschijnen. Prikt u daarna de soldeerklemmen van de potentiometer op de pennen 1A en 1B en u wijzigt de '3' in de PDL-functie in '1' op regel 30 dan moet u eveneens de getallen O t/m 255 zien. Ditzelfde geldt voor 7A/7B en 5A/5B. U zult wel het getal tussen haakjes in de PDL-functie moeten aanpassen.
In principe kunt u allerlei variabele weerstanden aan deze interface bevestigen; bijvoorbeeld lichtgevoelige weerstanden. Wanneer u zo'n lichtgevoelige weerstand aansluit en de lichtintensiteit varieert, kunt u met behulp van bovenstaand programma de waarde op het beeldscherm zien veranderen. U kunt dit doen door het licht van een lamp op de lichtgevoelige weerstand te laten vallen en daarna een hand tussen de lichtgevoelige weerstand en de lichtbron te houden. U zult zien dat de getallen op het scherm direct veranderen.
Suggestie: Met behulp van twee lichtgevoelige weerstanden en een stuk extra programmatuur moet het mogelijk zijn het LEGO-wagentje naar de lichtbron toe te laten rijden. Eenvoudig zal dit echter niet zijn.

4.5.1 De tekenhaak

Met behulp van interface 2 is het mogelijk een tekenmachine te bouwen, die in staat is eenvoudige tekeningen over te brengen naar het beeldscherm van de MSX-computer en daar te manipuleren. U zult wat moeten knutselen om het apparaat in elkaar te zetten. Maar voor het zover is, zal eerst de theoretische achtergrond worden belicht.

Wanneer u een tekening in de MSX-computer wilt overnemen, zult u van elk lijnstuk de XY-coodinaten moeten opgeven. Wat u eigenlijk doet, is over de tekening een assenstelsel leggen. De X-as ligt in de horizontale richting en de Y-as in de vertikale richting. Plotters die grafieken op papier weergeven, werken volgens dit principe. We zouden een lineaire schuifpotmeter voor de X-richting kunnen gebruiken en een lineaire schuifpotmeter voor de Y-richting. In de praktijk is dit haast niet te verwezelijken. Er is een andere manier. In winkels voor tekenmaterialen zijn zogenaamde pantografen te koop. Hiermee kunnen tekeningen worden vergroot en verkleind. In principe is een pantograaf te beschouwen als een schouder-elleboogconstructie. De twee draaipunten bestaan uit twee lineaire draaipotentiometers van 250 kilo-ohm; dezelfde als bij interface 2. De hoekverdraaiing van de potentiometers is door middel van de PDL-functie van de MSX-computer uit te lezen. Omdat ervan wordt uitgegaan dat de potentiometers lineair zijn, kan de hoek in graden worden bepaald. Het principe van het idee ziet u in afbeelding 4.5.5.



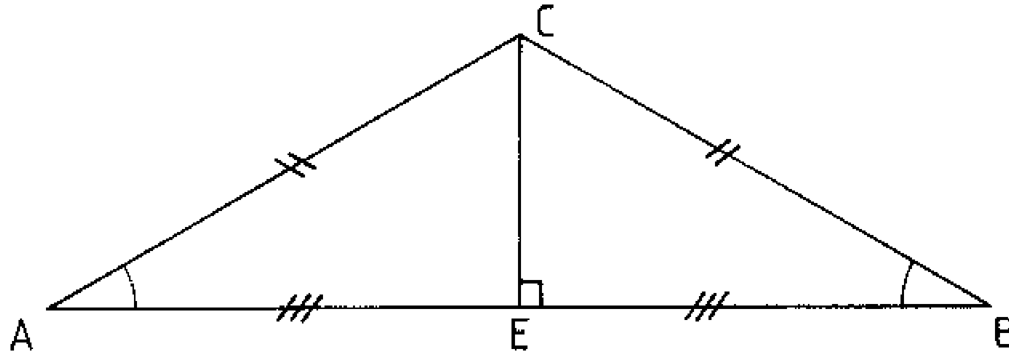
Afb. 4.5.5

Indien u eerst de tekenhaak wilt maken, kunt u het beste doorlezen bij de paragraaf die begint met 'constructie'.

Punt A is het aanwijspunt. Punt C en B zijn de draaipunten met de potentiometers. Verder zijn de lijnen BC en AC exact gelijk aan elkaar (20 cm). Er is dus sprake van een

gelijkbenige driehoek. De lengte van de basis van de driehoek (AB) verandert, wanneer het aanwijspunt wordt verplaatst. Het gaat er nu om, om van het aanwijspunt de XY-coo¶rdinaat te bepalen.

De driehoek ABC is een gelijkbenige driehoek. Zijde AC en BC zijn gelijk aan elkaar. Om de XY-coÃ¶rdinaat van het aangewezen punt A te berekenen, is het nodig om de lengte AB te kennen en de richting waarin AB ligt. Zijn de richting en de lengte bekend van de zijde AB dan kan door middel van eenvoudige goniometrische functies de XY-coordinaat worden berekend. Het komt er in het kort op neer dat polaire coo¶rdinaten naar Cartesiaanse coo¶rdinaten worden omgerekend.



Afb. 4.5.6

In afbeelding 4.5.6 ziet u weer dezelfde driehoek, maar nu is vanuit hoek C een loodlijn op de basis AB neergelaten. In een gelijkbenige driehoek geldt dat een loodlijn uit de top zowel de basis als de hoek C precies in tweeÃ«n deelt. Door het neerlaten van een loodlijn uit de tophoek C ontstaan twee rechthoekige driehoeken EBC en EAC, waarbij hoek E 90 graden is. De drie hoeken van een driehoek zijn samen 180 graden. Dit houdt in dat de halve hoek C uit de driehoek ABC plus hoek B samen 90 graden zijn. De grootte van hoek C is bekend; deze wordt bepaald door de hoekverdraaiing van potentiometer P1. Verder is de lengte van zijde CB bekend. Met behulp van wat goniometrisch rekenwerk (zie het programma 'Driehoek') kan AB worden berekend. Dit is tweemaal de lengte EB. Van de driehoek zijn nu alle gegevens bekend. De lengte van de zijden en de grootte van de hoeken. Het enige wat nog onbekend is, is de richting waarin AB ligt. Feitelijk wordt deze bepaald door de waarde van potentiometer P1 in het punt B. Hier zit nog wel een probleem. De hoekverdraaiing van potentiometer P1 is de hoek die wordt aangeduid met B' en niet hoek B''. Hoek B'' is te berekenen door bij hoek B', die wordt bepaald door de potentiometer P1 hoek B (zie afb 4.5.5) op te tellen. Het omzetten van de polaire coordinaten naar Cartesiaanse coo¶rdinaten (XY-coÃ¶rdinaten) geschiedt met de volgende formules:

X = lengte AB * COS B'' Y = lengte AB * SIN B''

Bij de berekening in de MSX-computer moet er rekening mee worden gehouden dat de BASIC-interpreter goniometrische functies in radialen berekent. Om met de digitizer te kunnen werken, zal deze moeten worden geijkt. Op het tekenbord worden twee punten aangewezen, waarvan de XY-coordinaten exact bekend zijn. Met behulp van de stelling van Pythagoras kan van ieder van de punten afzonderlijk de afstand worden bepaald tot

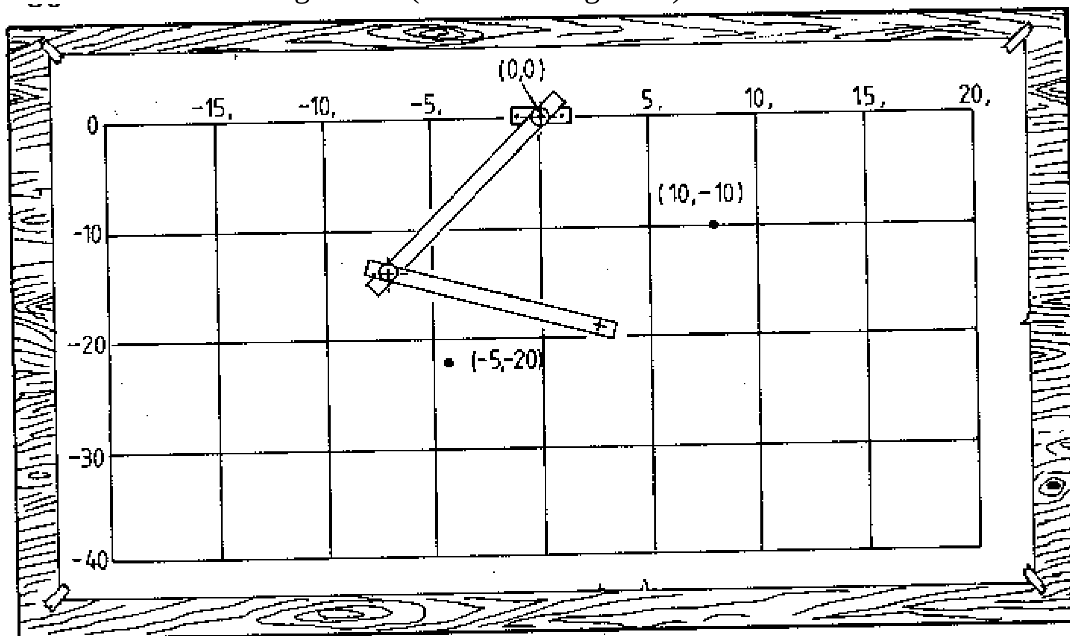
het 0,0 punt. De bijbehorende richtingen worden berekend met de formule: R = arctan(Y/X). Tegelijkertijd worden ook de waarden van beide potentiometers uitgelezen wanneer een punt wordt aangewezen. Er is uitgegaan van een lineair verband tussen de weerstandswaarde en de hoekverdraaiing. Dit betekent in formulevorm: G = aW + c.
G geeft het aantal graden aan. W is de waarde, die de PDL-functie oplevert. a is een factor waarmee de weerstandswaarde moet worden vermenigvuldigd en tenslotte is c een correctiefactor. Voor beide potentiometers moeten de waarden a en c worden bepaald.

In het programma 'Digitizer' zijn alle hierboven genoemde onderdelen terug te vinden.

De constructie:

Als eerste heeft u een tekenbord nodig van vijftig bij veertig centimeter. Aan de hand van de tekening kunt u zien hoe detekenhaak in elkaar zit en wat de maten zijn. Als materiaal voor de tekenhaak is 8 mm dik plexiglas-stafmateriaal gebruikt, maar u kunt even goed hardhouten latjes gebruiken. De lengte van beide benen van de tekenhaak is dertig centimeter. Met behulp van een verticale boorstandaard zijn de gaten geboord. De potentiometer zelf en de as van de potentiometer worden met M3 schroefjes vastgeklemd. Dit heeft het voordeel dat de potentiometers gemakkelijk in te stellen zijn. Het aanwijspunt is met behulp van een kraspen op het plexiglas aangegeven.

Op de tekenplank brengt u een assenkruis aan, waarbij het punt 0,0 ongeveer 2,5 centimeter van boven gemeten in het midden van de plank ligt. Slechts het derde en vierde kwadrant worden gebruikt (zie afbeelding 4.5.7).



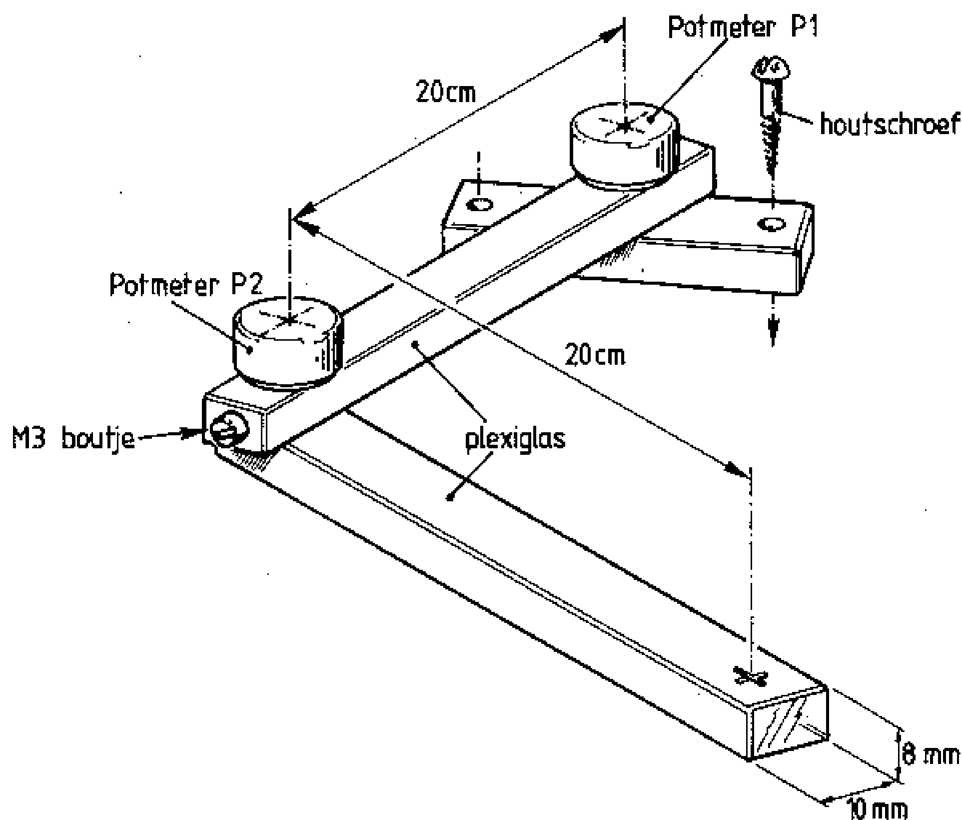
Afb. 4.5.7

Met behulp van passer en liniaal geeft u op de tekenplank het punt (10,-10) en (-5,-20) aan. U moet beide punten uitmeten in centimeters. Daarna bevestigt u het kleine stukje plexiglas (A) met behulp van twee houtschroeven op de tekenplank. De kern van het gat,

waarin de potentiometer komt, moet precies boven 0,0 liggen. Hierna kan de tekenhaak met de potentiometers worden bevestigd.

U moet er voor zorgen dat been B precies op de Y-as ligt en dat de potentiome-ter P1 ongeveer in de middenstand staat. Dit kunt u instellen met behulp van het kleine testprogramma bij interface 2. U moet ongeveer de waarde 127,128 op het beeldscherm zien. Bij potentiometer P2 doet u het zelfde. U zorgt ervoor dat been B op de Y-as ligt en been C evenwijdig aan de X-as ligt. Potentiometer P2 moet dan de waarde 127 a 128 opleveren. Staan de potentiometers in de goede stand dan kunnen zij door het aandraaien van de schroefjes worden vastgezet.

De tekenhaak samen met het bijbehorende programma zijn nu gereed voor gebruik.



Afb. 4.5.8

Het programma:

Wanneer u het programma start zal direct de ijkroutine worden gestart. In deze routine wordt u gevraagd om het aanwijspunt van de tekenhaak op punt A (10,-10) te plaatsen. U moet dit bevestigen door op de RETURN-toets te drukken. Hierna zal het programma vragen het aanwijspunt op punt B (-5,-20) te plaatsen. U bevestigt dit door op de RETURN-toets te drukken. Het programma heeft nu de tekenhaak geijkt en het menu verschijnt op het beeldscherm. U kunt met keuze 3 en 4 uit het menu een tekening op cassette plaatsen en terug lezen. Keuzemogelijkheid 1 geeft u de gelegenheid om een tekening over te nemen en op het beeldscherm te plaatsen. Het programma heeft nog een

aantal gegevens nodig, voordat kan worden begonnen met het overnemen van een tekening. U moet met het aanwijspunt de linkerbovenhoek en rechterbenedenhoek van de tekening aangeven, zodat de grootte van de tekening aan het programma bekend is. Hierna moet u aangeven waar op het scherm de linkerbovenhoek van de tekening moet komen te liggen. Ten slotte dient in het programma bekend te zijn hoe groot de tekening op het scherm moet worden afgebeeld. Dit wordt opgegeven door het aantal beeldpunten (pixels) in de breedte (X-richting) en in de hoogte (Y-richting) op te geven. De ingevoerde gegevens kunnen nadat een tekening is ingevoerd ook nog worden gewijzigd. Op deze manier kunt u tekeningen uitrekken of samenpersen. Nu de maten van een tekening bekend zijn aan het programma kan worden begonnen met het overnemen van de tekening. Hiervoor heeft u de cursortoet-sen en de functietoetsen 1 t/m 5 nodig. Op het beeldscherm verschijnt een zwart kruis. Wanneer u het aanwijspunt langzaam beweegt, beweegt het zwar te kruis over het beeldscherm in dezelfde richting mee. Drukt u op functietoets F1 dan wordt het zwarte kruis op het scherm vastgezet en er verschijnt direkt een tweede kruis. Verplaatst u het aanwijspunt naar een ander punt dan gaat het tweede kruis met de beweging mee. Zodra u vindt dat het tweede kruis ook op de goede plaats staat en u drukt F1 weer in dan zal tussen het eerste kruis en het tweede kruis een lijn worden getrokken. Het eerste kruis wordt gewist en het tweede zwarte kruis wordt het eerste kruis. Op die manier kunt u aaneengesloten kromme lijnen tekenen.

Wanneer u door middel van het aanwijspunt het kruis wilt verplaatsen zonder dat er een lijn wordt getrokken, moet u F2 indrukken. De kleur van het kruis verandert in wit. Wilt u weer een lijn gaan trekken, dan drukt u op F1. Met behulp van functietoets F5 onderbreekt u het tekenen en keert u terug naar het menu. Met functietoets F3 kunt u de laatst getrokken lijn wissen. Dit kan tot gevolg hebben dat in getrokken lijnen stukjes ontbreken. Dit komt, omdat de lijn wordt weggehaald door er in de achtergrondkleur een lijn overheen te trekken. Wanneer u F4 indrukt, wordt heel de tekening opnieuw getekend en zijn de onderbroken lijnen verdwenen.

In het programma vindt u alle hierboven genoemde onderdelen terug. Door de modulaire manier van opbouw van ook dit programma moet de werking te doorgronden zijn. Achteraan in het programma zit de stuurmodule, die het gehele programma onder controle houdt. Eerst wordt de module 'Initialiseren' gestart, zodat alle beginwaarden bekend zijn in het systeem en de sprite. Daarna volgt in een aparte module de ijkingsroutine. U moet er rekening meehou-den dat een MSX-computer niet met graden rekent, maar met radialen. Dit heeft tot gevolg dat er bij het omrekenen naar polaire coordinaten moet worden gecorrigeerd, omdat de coordinaten in het derde en vierde kwadrant liggen. In dezelfde routine wordt de factor waarmee de waarde uit de PDL-functie moeten worden vermenigvuldigd alsmede de constante berekend. Na deze vaste routine, komt het programma in het menu. Van hieruit kunnen de andere functies worden aangeroepen. In de routine 'X-Y paren' wordt de XY-coordi-naat berekend, waar het aanwijspunt van de tekenhaak zich bevindt. Het verdient zeker aanbeveling dit programma nauwkeurig te bestuderen.

```
10 REM *** DIGITIZER ***
20 KEY OFF
30 DIM H$(2),A(2,6),Z(300,4)
```

```
40 GOTO 2760
50 :
60 REM *** INITIALISEREN ***
70 PI = 4 * ATN(l)
80 R = 20       :REM lengte been
90 A(l,l)=10   : REM X-coord punt A
100 A(l,2)=-10 : REM Y-coord punt A
110 A(2,l)=-5  : REM X-coord punt B
120 A(2,2)=-20 : REM Y-COORD punt B
130 H$(l) = "[RETURN]"
14O H$(2) = "
150 FOR I=1 TO 10 : KEY I,"" : NEXT I
160 A$=STRING$(20,CHR$(0))
170 REM ** KRUIS / SPRITE **
180 SP$=LEFT$(A$,8)+CHR$(254)
190 SP$=SP$+LEFT$(A$,7)
200 SP$=SP$+STRING$(7,CHR$(128))
210 SP$=SP$+CHR$(63)+CHR$(O)
220 SP$=SP$+STRING$(7,CHR$(128))
230 DEF FNA(A) = INT(A +.5)
240 RETURN
250 :
260 REM *** KOP ***
270 SCREEN 0 : COLOR 15,4,4 : CLS
280 PRINT "*** TEKENHAAK ***"
290 PRINT F$
300 RETURN
310 :
320 REM *** INVOER ***
330 H=0 : J=l : LOCATE 2,Z
340 PRINT A$ : PRINT "  ";B$
350 LOCATE 2,Z+3
360 PRINT "Druk daarna op ";H$(J)
370 H=H+1 : IF H < 10 THEN 390
380 J=J+1 : H=0 : IF J = 3 THEN J=l
390 C$=INKEY$ : IF C$="" THEN 350
400 P1 = 0 : P2 = 0
410 FOR I = l TO 3
420 IF C$ <> CHR$(13) THEN 350
430 P1 = P1+PDL(1) : P2 = P2+PDL(3)
440 NEXT
450 P1 = Pl/3 : P2 - P2/3
460 RETURN
470 :
480 REM *** IJKEN ***
490 F$ = "- IJkroutine -"
500 GOSUB 270   :REM *** KOP
510 A$ = "Zet tekenhaak op "
520 B$ = "punt A (10,-10)" : Z=5
530 GOSUB 330   :REM *** INVOER
540 A(1,5)=P1 : A(1,6)=P2
550 B$ = "punt B (-5,-20)" : Z=ll
560 GOSUB 330   :REM *** INVOER
570 A(2,5) = P1 : A(2,6) = P2
580 IF P1 <> P2 THEN 630
590 CLS : BEEP
600 LOCATE 2,10
```

```
610 PRINT "tekenhaak niet goed !"
620 STOP
630 GOSUB 670   :REM *** IJKINGSBER.
640 RETURN
650 :
660 REM *** IJKINGSBEREKENING ***
670 FOR I = l TO 2
680 Z = SQR(A(I,1)^2 + A(I,2)^2)
690 A = Z / (2*R)
700 A = 2 * ATN(A/SQR(1-A*A))
710 A(I,4) = A
720 B=ATN(A(I,2)/A(1,1)
730 IF A(I,1)< 0 AND A(I,2)>=0 THEN B=B+P1
740 IF A(I,1< 0 AND A(I,2) < 0 THEN B=B+P1
750 IF A(I,1)>0 AND A(I,2< 0 THEN B=B+2*P1
760 A(I,,3)=B
770 A(I,3) = A(I,3)- .5*(P1 - A)
780 NEXT I
790 REM ** FACTOR/CONSTANTE **
800 FOR I = l TO 2
810 D = A(I,2+1) - A(2,2+I)
820 P = A(I,4+1) - A(2,4+I)
830 Fd) = D/P
840 Cd) = A(I,2+I)-F(I)*A(1,4+I)
850 NEXT I
860 RETURN
870 :
880 REM *** ABSOLUTE GROOTTE ***
890 F$ = "-Grootte bepaling-"
900 D$ = " van de tekening."
910 E$ = "Zet de tekenhaak in de "
920 GOSUB 270   :REM *** KOP
930 A$ = E$ + "linker-"
940 B$ = "boven hoek"+D$
950 Z = 5 : GOSUB 330
960 GOSUB 1400   :REM *** X-Y PAREN
970 A(1,1) = X : A(2,1) = Y
980 A$ = E$ + "rechter-"
990 B$ = "beneden hoek"+D$
1000 Z = 11: GOSUB 330
1010 GOSUB 1400   :REM *** X-Y PAREN
1020 A(l,2) = X : A(2,2) = Y
1030 TX = A(1,l)*-l : TY = A(2,l)*-l
1040 GX = ABS(A(1,2)-A(1,1))
1050 GY =ABS(A(2,2)-A(2,1))
1060 RETURN
1070 :
1080 REM *** PLAATS OP SCHERM ***
1090 F$ = "-plaats & grootte-"
1100 GOSUB 270
1110 LOCATE 0,5
1120 PRINT "Geef het X-Y coordinaat,"
1130 PRINT "waar de linkerbovenhoek"
1140 PRINT "moet komen ligqen."
1150 PRINT "Geef X-coordinaat ";
1160 INPUT A(1,3)
1170 PRINT
```

```
1180 PRINT "Geef Y-cordinaat  ";
1190 INPUT A(2,3)
1200 LOCATE 0,12
1210 PRINT "Geef de breedte van de"
1220 PRINT "af te beelden tekening in'
1230 PRINT "pixels <0-255> ";
1240 INPUT A(1,4)
1250 PRINT
1260 PRINT "Geef de hoogte van de "
1270 PRINT "af te beelden tekening in'
1280 PRINT "pixels <0-190> ";
1290 INPUT A(2,4)
1300 LOCATE 0,20
1310 PRINT "Invoer accoord <J/N>  "
1320 LOCATE 20,20 : A$ = INPUT$(1)
1330 H=INSTR("JjNn",A$)
1340 IF H = O THEN 1300
1350 IF H > 2 THEN 1100
1360 FX = A(1,4)/GX : FY = A(2,4)/GY
1370 RETURN
1380 :
1390 REM *** X-Y PAREN ***
1400 P1 = 0 : P2 = 0
1410 FOR I = l TO 2
1420 P1=P1+PDL(1) : P2=P2+PDL(3)
1430 NEXT I
1440 P1 = Pl/2 : P2 = P2/2
1450 B = F(l) * P1 + C(1)
1460 C = F(2) * P2 + C(2)
1470 Z = SQR(2*R^2-2*R*R*COS(C)
1480 B1= (P1 - C)/2 + B
1490 X = INT(Z*COS(BI)*100+.5)/100
1500 Y = INT(Z*SIN(BI)*100+.5)/100
1510 RETURN
1520 :
1530 REM *** CASSETTE SCHRIJVEN ***
1540 F$ = "-cassette schrijven-"
1550 GOSUB 270  :REM *** KOP
1560 OPEN "CAS:" FOR OUTPUT AS #1
1570 PRINT #1,N;TX;TY;GX;GY
1580 FOR I = l TO N
1590 PRINT #1,Z(I,1);Z(I,2)
1600 PRINT #1,Z(I,3);Z(I,4)
1610 NEXT I
1620 CLOSE #1
1630 RETURN
1640 :
1650 REM *** CASSETTE LEZEN ***
1660 F$ = "-cassette lezen-"
1670 GOSUB 270  :REM *** KOP
1680 OPEN "CAS:" FOR INPUT AS #1
1690 INPUT #1,N,TX,TY,GX,GY
1700 FOR I = l TO N
1710 INPUT #1,Z(I,1),Z(I,2)
1720 INPUT #1,Z(I,3),Z(I,4)
1730 NEXT I
1740 CLOSE #1
```

```
1750 RETURN
1760 :
1770 REM *** KEY l ***
1780 REM * LIJN TREKKEN *
1790 IF KI = 0 THEN 1840
1800 LINE(X3,Y3)-(PX,PY),1
1810 N = N + l   : KI = 0
1820 Z(N,1) = X4 : Z(N,2) = Y4
1830 Z(N,3) = X2 : Z(N,4) = Y2
1840 X3 = PX : Y3 = PY
1850 X4 = X2 : Y4 = Y2
1860 PUT SPRITE 2,(X3-8,Y3-8),1,1
1870 K1 = K1 + l : KL = l : L = l
1880 RETURN
1890 :
1900 REM *** KEY 2 ***
1910 REM * VERPLAATSEN  *
1920 PUT SPRITE 2,(255,190),l,0
1930 K1 = 0 : KL = 15 : L = 2
1940 RETURN
1950 :
1960 REM *** KEY 3 ***
1970 RE * LAATSTE LIJN WISSEN *
1980 GOSUB 1920 :REM *** KEY 2
1990 IF N = 0 THEN 2090
2000 REM ** WISROUTINE **
2010 X3 = Z(N,1) : Y3 = Z(N,2)
2020 X4 = Z(N,3) : Y4 = Z(N,4)
2030 X3 = INT(A(l,3)+X3*FX+.5)
2040 X4 = INT(A(l,3)+X4*FX+.5)
2050 Y3 = INT(A(2,3)+Y3*FY+.5)
2060 Y4 = INT(A(2,3)+Y4*FY+.5)
2070 LINE (X3,Y3)-(X4,Y4),0
2080 N = N - l : L = 3
2090 RETURN
2100 :
2110 REM *** KEY 5 ***
2120 REM * ROUTINE VERLATEN *
2130 L = 4
2140 RETURN
2150 :
2160 REM *** KEY 4 ***
2170 REM ** TEKENING OP SCHERM **
2180 IF N <= 0 THEN 2280
2190 FOR I = l TO N
2200 X3 = Z(I,1) : Y3 = Z(I,2)
2210 X4 = Z(I,3) : Y4 = Z(I,4)
2220 X3 = INT(A(l,3)+X3*FX+.5)
2230 X4 - INT(A(l,3)+X4*FX+.5)
2240 Y3 = INT(A(2,3)+Y3*FY+.5)
2250 Y4 = INT(A(2,3)+Y4*FY+.5)
2260 LINE (X3,Y3)-(X4,Y4),1
2270 NEXT I
2280 RETURN
2290 :
2300 REM *** OPBOUW ***
2310 IF N > 0 THEN 2340
```

```
2320 GOSUB 890  :REM *** ABS GROOTTE
2330 GOSUB 1090 :REM *** PLAATS
2340 COLOR 15,4,4 : SCREEN 2,2 : CLS
2350 GOSUB 2180 : SPRITE$(1)=SP$
2360 L = 3 : KL = 15 : K1 = 0
2370 ON KEY GOSUB 1790,1920,1980,2180,2120
2380 KEY(l) ON : KEY(2) ON
2390 KEY(3) ON : KEY(4) ON
2400 KEY(5) ON
2410 GOSUB 1400 :REM *** X-Y PAREN
2420 X2 = X + TX : Y2 = ABS(Y + TY)
2430 PX=INT(A(l,3)+X2*FX+.5)
2440 PY=INT(A(2,3)+Y2*FY+.5)
2450 PUT SPRITE 1,(PX-8,PY-8),KL,1
2460 IF L < 4 THEN 2410
2470 KEY(l) OFF : KEY(2) OFF
2480 KEY(3) OFF : KEY(4) OFF
2490 KEY(5) OFF
2500 RETURN
2510 :
2520 REM *** MENU ***
2530 F$ = "-menu-" : GOSUB 27O
2540 LOCATE 2,5
2550 PRINT "0 = Stoppen"
2560 LOCATE 2,7
2570 PRINT "l = Tekening overnemen"
2580 LOCATE 2,9
2590 PRINT "2 = Plaats op het scherm"
2600 LOCATE 2,11
2610 PRINT "3 = Inlezen van cassette"
2620 LOCATE 2,13
2630 PRINT "4 = Schrijven naar cas."
2640 LOCATE 0,18
2650 PRINT STRING$(37,"-")
2660 LOCATE 2,20
2670 PRINT "Wat is uw keuze ?   ";
2680 LOCATE 20,20 : A$ =INPUT$(1)
2690 PRINT A$
2700 M = 1NSTR("01234",A$)-1
2710 IF M = -l THEN 2660
2720 IF M=2 AND N=0 THEN 2530
2730 RETURN
2740 :
2750 REM **** STUURMODULE ****
2760 GOSUB 70   :REM *** INIT
2770 GOSUB 490  :REM *** INVOER IJK
2780 GOSUB 2530 :REM *** MENU
2790 IF M = l THEN GOSUB 2310
2800 IF M = 2 THEN GOSUB 1090
2810 IF M = 3 THEN GOSUB 1660
2820 IF M = 4 THEN GOSUB 1540
2830 IF M <> 0 THEN 2780
2840 CLS
2850 END
```

Daarbij zijn de Basic programma’s getest met het in het volgende beschreven checksum programma. De in het boek opgenomen programma bevatte een vervelende bug, hier gecorrigeerd!

Appendix

Met behulp van het hier beschreven programma kunnen de programma's uit dit boek worden gecontroleerd op fouten. Het programma controleert op drie punten:

- Elke BASIC-regel is niet langer dan vijftig posities. Zijn er meer dan vijftig posities dan verschijnt een foutmelding op het beeldscherm.

- Elk programma begint met regelnummer 10. De regelnummers lopen met 10 op. Er verschijnt een foutmelding op het beeldscherm als de volgorde niet meer klopt.

- Wanneer u de listings uit dit boek bekijkt, ziet u rechts op elke BASIC-regel een schuine streep met daarachter een controlegetal. Dit controlegetal of checksum moet overeenstemmen met het getal dat dit programma berekent. Zijn de getallen gelijk aan elkaar dan is de BASIC-regel zeer waarschijnlijk correct ingetoetst. Stemmen de getallen niet overeen dan zit er een fout in de betreffende BASIC-regel.

Denk erom: u moet bij het intikken van de programma's de schuine streep met

het controlegetal NIET overnemen.

Bij het intikken van de programma's mag u de spaties tussen de statements of variabelen weglaten. U mag de spaties die tussen aanhalingstekens voorkomen echter niet verwijderen. De REM-statements aan het begin van BASIC-regels moeten worden ingetikt. De REM-statements achter een dubbelepunt mogen worden weggelaten.

Het programma 'Checksum' maakt gebruik van een eenvoudig controlemechanisme. De ASCII-waarden van de karakters op een BASIC-regel worden bij elkaar en bij het regelnummer opgeteld. Dit controlegetal wordt samen met het bijbehorende regelnummer op het scherm afgedrukt. Om met dit programma te kunnen werken, moet u de volgende procedure volgen:

- U toetst eerst een compleet programma in. Hierna controleert u of het programma werkt. U verbetert de direct herkenbare typfouten. Kunt u verder geen fouten vinden, sla het programma dan op cassette op met het CSAVE-statement.

- Schrijf het ingetoetste programma nogmaals op een andere casssette weg met het SAVE-statement, gevolgd door een programmanaam, een komma en een A. Als opdracht ziet dit er als volgt uit:

```
SAVE "programmanaam",A
```

Door het op deze manier wegschrijven van een programma wordt een ASCII-bestand naar cassette geschreven. Dit duurt vrij lang! Het programma 'checksum' kan dit bestand teruglezen.

U laadt nu het programma 'Checksum' en spoelt de cassette die u net heeft aangemaakt terug naar het begin.

Het programma vraagt of de cassetterecorder aanstaat. Wanneer u deze vraag bevestigend beantwoordt, begint het programma het ASCII-bestand regel voor regel te lezen. Indien de lengte van de BASIC-regels korter is dan vijftig posities en de regelnummering met tien oploopt, verschijnt geen foutmelding.

Op het beeldscherm verschijnen de regelnummers van het als ASCII-bestand weggeschreven programma met daarachter de checksums. Deze checksums of controlegetallen worden door het checksumprogramma bepaald. De controlegetallen uit het checksumprogramma moeten overeenstemmen met de getallen rechts achter de schuine streep in de listing. ledere keer verschijnen vijftien regelnummers met bijbehorende controlegetallen op het scherm.

Dit programma kunt u ook met een diskettestation gebruiken. U dient dan wel een aantal wijzigingen aan te brengen:

u toetst het programma 'Checksum' met uitzondering van de regels 140,150 en 160 in. Regel 170 wijzigt u in:

170 OPEN "LYST.TXT" FOR INPUT AS #1

Na deze wijzigingen kunt u het programma gebruiken. Wanneer u een programma heeft ingetoetst en de meest voor de hand liggende typfouten heeft gecorrigeerd, stelt u het programma veilig op diskette met behulp van het commando:

SAVE "programmanaam"

Hierna schrijft u het programma nogmaals weg, maar nu als een ASCII-bestand met de opdracht:

SAVE "lyst.txt",A

Nadat het programma als LYST.TXT is weggeschreven, start u het checksumprogramma. Dit programma zal nu automatisch het bestand LYST.TXT oppakken. De verdere werking van het checksumprogramma blijft ongewijzigd.

```
10 REM *** CHECKSUM ***
20 CLEAR 5000
30 KEY OFF
40 GOTO 1130
50 :
60 :REM *** INITIALIEREN ***
```

```
70 RG = 0       :REM regelteller
80 TL = 0       :REM afdruk regels
90 H$ =  STRING$(37,CHR$(45))
100 I$ = "regel ######   checksum"
110 I$ = I$ + "########"
120 F$ = ""
130 GOSUB 240
140 LOCATE 2,20
150 INPUT "Welke file";F$
160 IF F$ = "" THEN 1290
170 OPEN F$ FOR INPUT AS#1
180 SCREEN 0
190 COLOR 15,4,4
200 DEF USR1=&H156 :REM keybrd leeg
210 RETURN
220 :
230 REM *** KOP ***
240 CLS        :REM scherm wissen
250 PRINT "*** CHECKSUM ***"
260 PRINT F$
270 RETURN
280 :
290 REM *** DOORGAAN ***
300 LOCATE 2,22
310 DUMMY = USR1(0)
320 PRINT "Doorgaan [Ja/Nee] ?     "
330 LOCATE 22,22 : A$ = INPUT$(1)
340 PRINT A$ : S = INSTR("JjNn",A$)
350 IF S = 0 THEN 300
360 S = INT(S/2 + .5)
370 RETURN
380 :
390 REM  *** REGEL LEZEN ***
400 FF = 0
410 IF EOF(1)=-1 THEN EF=1 ELSE EF=0
420 IF EF = 1 THEN 470
430 LINE INPUT#1,Z$  : L = LEN(Z$)
440 IF L < 50 THEN 470
450 PRINT " Regel is te lang."
460 PRINT Z$ : FF = 1
470 RETURN
480 :
490 REM *** REGELNUMMER ***
500 I= 0 : FF = 1
510 I =  I + 1
520 IF I > L THEN 580
530 A$ = MID$(Z$,I,1)
540 IF A$ <>  CHR$(32) THEN 510
550 RB = VAL(LEFT$(Z$,I)): RG=RG+10
560 IF RB = RG THEN FF=0 ELSE FF=1
570 IF FF = 0 THEN 600
580 PRINT "Regelnr moet";RG;"zijn."
590 PRINT Z$
600 L = L-I : Z$=RIGHT$(Z$,L)
610 IF TL <> 0 THEN 640
620 L1 = LEN(Z$)-4
630 F$ = RIGHT$(Z$,L1)
```

```
640 RETURN
650 :
660 REM *** REMark ***
670 E$ = MID$(Z$,X,3) : RM = 0
680 IF E$="REM" OR E$="rem" THEN 690 ELSE 700
690 CH = CH + 228 : RM = 1
700 RETURN
710 :
720 REM *** SPATIES VERWIJDEREN ***
730 Q$ = "" : H=0
740 FOR I = 1 TO L
750 A$ = MID$(Z$,I,1)
760 IF A$ <> CHR$(34) THEN 780
770 IF H = 0 THEN H=1 ELSE H=0
780 IF A$ <> CHR$(32) THEN 800
790 IF H = 0 THEN A$=""
800 Q$ = Q$ + A$
810 NEXT I
820 L = LEN(Q$) : Z$ = Q$
830 RETURN
840 :
850 REM *** SCAN ***
860 CH = 0
870 FOR I = 1 TO L
880 A$ = MID$(Z$,I,1)
890 IF A$ <> CHR$(58) THEN 940
900 IF H = 1 THEN 940
910 X = I+1 : IF X+3 > L THEN 940
920 GOSUB 670   :REM *** REMark
930 IF RM=1 THEN I=L : CH=CH-286
940 CH = CH + ASC(A$)
950 NEXT I
960 CH = CH + RG
970 RETURN
980 :
990 REM *** AFDRUK ***
1000 S = 1
1010 IF TL MOD 15 <> 0 THEN 1080
1020 IF TL = 0 THEN 1060
1030 LOCATE 0,19 : PRINT H$
1040 GOSUB 300   :REM *** DOORGAAN
1050 IF S = 2 THEN 1100
1060 GOSUB 240   : REM *** KOP
1070 LOCATE 0,3
1080 PRINT USING I$;RG;CH
1090 TL = TL + 1
1100 RETURN
1110 :
1120 REM **** STUURMODULE ****
1130 GOSUB 70    : REM *** INIT
1140 GOSUB 240   : REM *** KOP
1150 GOSUB 400   : REM *** REGEL LEZEN
1160 IF EF = 0 THEN 1180
1170 PRINT "Einde file !" : FF= 1
1180 IF FF = 1 THEN 1280
1190 GOSUB 500  : REM *** REGELNUMMER
1200 IF FF = 1 THEN 1280
```

```
1210 X = 1
1220 GOSUB 670  :REM *** REMark
1230 IF RM = 1 THEN 1260
1240 GOSUB 730  : REM *** SPATIE VERW
1250 GOSUB 860  : REM *** SCAN
1260 GOSUB 1000  :REM *** AFDRUK
1270 IF S = 1 THEN 1150
1280 CLOSE#1
1290 END
```